

CONNECTED HOME

Memoria del proyecto

Autor: Alberto Soler Rodríguez

Director: Juan Belmonte Rodríguez

Ponente: Marc Alier Forment

Ingeniería del Software

23 – 06 – 2016

ÍNDICE DE CONTENIDO

1. Contexto

1.1 Introducción.....	5
1.2 Actores implicados	6

2. Estado del arte

2.1 Contextualización	7
2.2 Estudio del mercado	8
2.3 Conclusiones del estudio de mercado.....	10

3. Formulación del problema

11

3.1 Objetivo principal.....	12
3.2 Objetivos específicos	13

4. Alcance

14

4.1 Posibles obstáculos.....	16
------------------------------	----

5. Metodología y rigor

5.1 Métodos de trabajo	17
5.2 Herramientas de seguimiento	18
5.3 Métodos de validación.....	19

6. Planificación temporal

6.1. Planificación general	20
6.2 Descripción de las tareas	23
6.3 Calendario	25

7. Gestión económica

7.1. Consideraciones iniciales	26
7.2. Identificación y estimación de costes	27
7.3. Control de gestión.....	32

8. Sostenibilidad y compromiso social

8.1. Económica	33
8.2. Social	34
8.3. Ambiental	35

9. Descripción general de los requisitos

9.1 Requisitos funcionales.....	36
9.2 Requisitos no funcionales	39
9.3 Requisitos de datos.....	47

10. Especificación

10.1 Actores.....	48
10.2 Diagrama de contexto	49
10.3 Diagramas de historias de usuario	50
10.4 Especificación textual de las historias de usuario.....	52
10.5 Modelo de comportamiento.....	62

11. Diseño del servidor	76
12. Diseño de los módulos Arduino	83
13. Aplicación web	
13.1 Diseño de las pantallas	95
13.2 Interacción con el usuario.....	97
14. Aplicación móvil	
14.1 Diseño de las pantallas	100
14.2 Interacción con el usuario.....	104
15. Interacción entre componentes del sistema	110
16. Conclusiones	113
17. Referencias	115
18. Bibliografía	116
Anexo 1. Diagrama de Gantt	117
Anexo 2. Versiones de Android y iOS	118

ÍNDICE DE TABLAS

Tabla 1. Recursos materiales.....	21
Tabla 2. Estimación de horas.....	25
Tabla 3. Costes recursos humanos.....	27
Tabla 4. Costes directos por actividad.....	28
Tabla 5. Costes indirectos.....	29
Tabla 6. Costes de contingencia.....	30
Tabla 7. Costes de imprevistos.....	30
Tabla 8. Presupuesto.....	31
Tabla 9. Valoración de sostenibilidad.....	33

ÍNDICE DE FIGURAS

Fig. 1 Esperanza de vida en España.....	7
Fig. 2 Evolución de la población mayor, 1900-2061.....	11
Fig. 3 Fases de la metodología en cascada.....	17
Fig. 4 Diagrama de contexto	49
Fig. 5 Diagrama del sistema	50
Fig. 6 Diagrama de la aplicación móvil	51

Fig. 7 Diagrama de la aplicación web	51
Fig. 8 Raspberry Pi	76
Fig. 9 Conexiones de la Raspberry Pi	78
Fig. 10 Conexiones del protocolo Mosquitto	79
Fig. 11 Mapa de las conexiones MQTT del sistema	80
Fig. 12 Arquitectura de opoenHAB	81
Fig. 13 Comunicaciones de opoenHAB	81
Fig. 14 Arduino UNO R3	83
Fig. 15 Conexiones y pines de la placa Arduino UNO R3	84
Fig. 16 ZigBee shield	86
Fig. 17 Montaje de la ZigBee shield en la placa Arduino	86
Fig. 18 Módulo Xbee Serie 2	87
Fig. 19 Módulo ESP8266	88
Fig. 20 UART	89
Fig. 21 Leds	89
Fig. 22 Zumbador	89
Fig. 23 Pulsador	89
Fig. 24 Circuito del módulo Wifi	90
Fig. 25 Sensor de Gas	91
Fig. 26 Circuito del módulo de Gas	91
Fig. 27 Sensor de llama	92
Fig. 28 Circuito del módulo de fuego	92
Fig. 29 Sensor magnético	93
Fig. 30 Circuito del módulo de Puerta Abierta	93
Fig. 31 Sensor de humedad	94
Fig. 32 Circuito del módulo de Inundación	94
Fig. 33 Diagrama de componentes	110
Fig. 34 Diagrama de conexiones	111
Fig. 35 Diagrama de software	112

1. CONTEXTO

1.1 Introducción

Este proyecto es el Trabajo Final de Grado de la especialidad de Ingeniería del Software de la Facultad de Informática de Barcelona (Universidad Politécnica de Cataluña) ^[1]. Se trata de un proyecto de modalidad B, en el cual la empresa para la que trabajo, *Worldline* ^[2], está desarrollando un proyecto a gran escala llamado ‘*Connected Assistance*’.

Éste proyecto consta varias aplicaciones que utilizan el Internet de las cosas, conocido como ‘*Internet of Things*’ ^[3], para ayudar a personas que necesitan una atención especial o un seguimiento. No solo les permite el seguimiento y la atención personalizada de su salud sino que aumenta su calidad de vida ayudándoles o guiándoles en sus tareas u obligaciones.

Mi aportación para éste proyecto será montar un conjunto de detectores y actuadores para controlar diferentes peligros que puedan haber en una casa donde viva una persona con diferentes necesidades, y controlarlas mediante una aplicación que desarrollaré para un teléfono móvil.

1.2 Actores implicados

En este apartado se definirán los principales actores (*Stakeholders*) en este proyecto, es decir, a quien va dirigido el producto, quien lo hará servir... en definitiva, todas las personas que tengan algún tipo de relación con el proyecto.

Paciente

El principal actor implicado será el cliente, que será cualquier persona que necesite un seguimiento y una atención especial personalizada y que contrate nuestros servicios.

Familiar del paciente

Los pacientes para los cuales trabajamos tendrán que tener alguna persona responsable a la cual acudir en caso de alguna emergencia o con la que contactar para avisar o alertar de cualquier noticia o peligro relacionado con el paciente.

Empresas externas

Habrà empresas que estarán directamente implicadas en nuestro proyecto, como por ejemplo alguna mutua o la misma seguridad social, ya que podrán llevar un control y un seguimiento de todos sus pacientes que tengan nuestro sistema contratado.

Worldline

Worldline es la empresa para la cual estoy desarrollando parte de éste gran proyecto, por lo tanto es otro actor implicado, ya que será el que utilizará mi software para su proyecto final.

2. ESTADO DEL ARTE

2.1 Contextualización

Cada vez la esperanza de vida es mayor, y por eso hay que trabajar en mejorar el apoyo de la asistencia sanitaria para poder dar una mayor calidad de vida a todas las personas mayores y/o vulnerables y a sus familias. Esta situación preocupa a todo el mundo, y sobre todo a gobiernos que tienen que buscar y promover soluciones más eficientes para los próximos años.

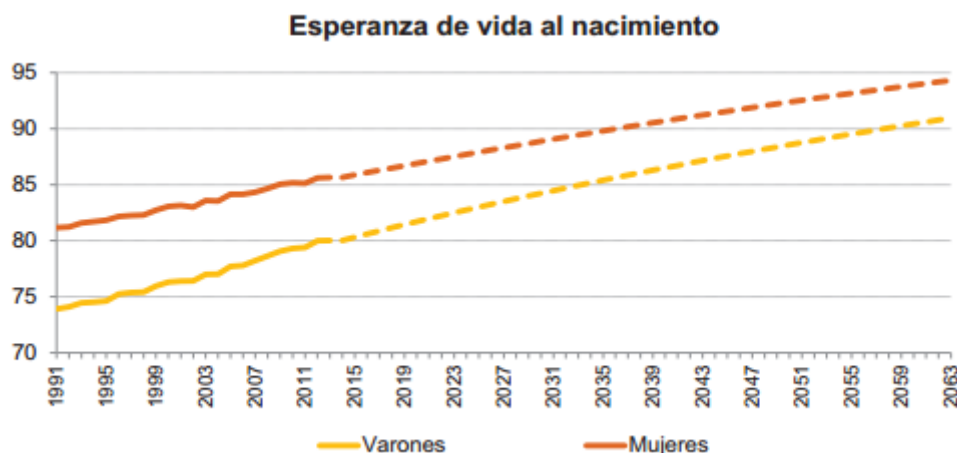


Fig. 1 Esperanza de vida en España

Worldline Connected Assistance, el proyecto que engloba a *Connected Home*, tiene como objetivo resolver buena parte de este problema, ofreciendo servicios para la gestión de la vida de las personas que necesitan una atención especial centrados en la tecnología móvil. Se pretende crear una Red de Asistencia que puede ayudar a los pacientes a vivir de forma independiente, al tiempo que ofrece seguridad y tranquilidad a los familiares.

Los pacientes estarán monitorizados y atendidos a través de diferentes dispositivos y periféricos, que permitirán monitorizar sus constantes vitales. Estos datos serán analizados de forma continuada por el personal médico y sanitario, para actuar inmediatamente en caso de necesidad. Estos datos estarán recogidos de manera segura en la nube, y se podrán consultar y analizar en tiempo real, dando la posibilidad a los servicios médicos de seguir el progreso de los pacientes.

El objetivo es aportar seguridad y bienestar, simplificando el manejo de la salud de los pacientes por parte de los propios pacientes junto con sus familiares y los servicios médicos.

Connected Assistance comparte la información con los familiares elegidos para ayudarles a ser proactivos en la detección de posibles problemas. Éstos familiares también podrán obtener información médica y datos del paciente a través de un portal Web.

Connected Home será una parte de la solución ofrecida por *Connected Assistance*. Ofrecerá un conjunto de dispositivos, que se podrán adaptar o escoger según las necesidades de los pacientes, para facilitar y hacer más seguro el hogar donde resida el paciente.

Como estará dentro del *Connected Assistance*, contará con todas sus facilidades, como el poder consultar datos a tiempo real a partir de un portal Web o una aplicación móvil.

2.2 Estudio del mercado

El mercado al que está enfocado *Connected Home* es un mercado que va en aumento, como ya se ha demostrado en la Fig.1 del apartado anterior. El aumento de la esperanza de vida hace que nos tengamos que preocupar más por las personas que necesitan una atención continuada.

Junto la esperanza de vida global, también ha aumentado la esperanza de vida dentro de las personas que necesitan una atención especial, haciendo que sean el doble de vulnerables: por una parte debido a su discapacidad y por otra parte por la edad.

El problema que tienen la mayor parte de las personas que tienen que estar pendientes de una persona mayor o con discapacidad, es que no tienen capacidad para estar pendientes de esa persona tanto tiempo como quisieran debido a sus propias obligaciones.

La mejor solución para estas personas es tener una aplicación desde donde puedan consultar, en tiempo real, los datos de la persona de la cual tienen que estar pendientes. Y no solo los datos de salud de esa persona, sino también datos de seguridad como saber si esa persona está en casa o saber si se ha dejado la puerta abierta,...

Antes de sumergirse de lleno en crear una solución a todas estas necesidades, hemos buscado las empresas y los productos que ofrece actualmente el mercado, para saber si merece la pena desarrollar un nuevo producto para poder dar solución a este problema.

Estas son algunas de esas empresas:

Lively

Se trata de pequeños sensores que se colocan dentro de objetos que hay en la casa – en frascos de pastillas, o el frigorífico – y detectan cuando el residente está tomando los medicamentos, ingiriendo alimentos, o saliendo de la casa. Las señales de actividad se envían desde los sensores a la página web de *Lively* (no requiere conexión *WiFi*), donde se recoge la información para los familiares y cuidadores. Los datos también se comparten a través de *Smartphone* y notificaciones de correo electrónico, en caso de producirse cualquier actividad irregular. Dos veces al mes, *LivelyGram* genera automáticamente un impreso con fotos y mensajes de los familiares y amigos que tienen acceso. ^[4]

PocketFinder

Estos discretos dispositivos permiten a las familias tener la tranquilidad de poder controlar la movilidad de sus seres queridos a través de un localizador GPS al que se mantienen conectados. Estos localizadores GPS proporcionan información de la localización en tiempo real, y dicha actividad puede seguirse a través de un portal web o *Smartphone*. También existe un dispositivo *PocketFinder* que se instala directamente en el vehículo y funciona con la batería del coche. ^[5]

Guardian Medical Monitoring

Guardian ofrece un sistema virtual de asistencia y vigilancia que permite a los miembros de la familia realizar un seguimiento de sus seres queridos que viven de forma independiente a través de una cámara que transmite una comunicación visual y audible. Lo cual disminuye la necesidad de tener que gastar dinero en personal de asistencia, reduciendo los costes en cuidadores o bien en la cantidad de horas de atención en centros de ancianos. ^[6]

MC10's BioSTAMP

Aunque no ha llegado aún al mercado (en 3-5 años), el *BioSTAMP* cambia el concepto de diagnóstico médico para todas las edades. Aplicado como una tirita o tatuaje temporal, el bio sello mide todos los niveles de hidratación (fundamental para la tercera edad), la temperatura corporal, la frecuencia cardiaca, la actividad cerebral, e incluso la exposición a los rayos UV. Se trata de una tecnología inalámbrica que permite cargar los datos en un *Smartphone* próximo para su análisis (por ejemplo, ¡el médico puede comprobar el estado de la abuela sin que ésta tenga que desplazarse!).^[7]

Independa y LG

Un interfaz cuidadosamente diseñado conocido como “*Angela*” es Independa, con reconocimiento de voz e integrado en pantalla de gran tamaño, listo para usarse con fuentes de pantalla de mayor tamaño y contraste para las personas mayores. Cuando se activa, es posible navegar por Internet, utilizar video chat con amigos y familiares, acceso simple a través de email, juegos, ver fotos familiares, seguir un horario diario, recibir notificaciones para recordar la medicación y más. *Angela* incluso puede programarse para llamar a mamá o papá y recordarles que deben tomar sus medicamentos, y todos estos servicios están disponibles a través de televisión y portátil, en casas particulares donde vivan personas mayores y también en centros especializados en asistencia a la tercera edad.^[8]

Teleasistencia

Se trata de un servicio público que pretende ayudar a la persona mayor a mejorar su calidad de vida en un doble aspecto: social y sanitario. La finalidad es evitar situaciones de soledad, abandono, de inseguridad, proporcionando una comunicación continua con el domicilio de la persona para poder responder con inmediatez en caso de emergencia. Es un dispositivo de fácil manejo que pone en contacto al usuario con un Centro de Atención compuesto por profesionales preparados para responder ante cualquier tipo de incidencia. El dispositivo consta de:

- un pulsador personal con forma de pulsera o colgante, fácil de usar, que los usuarios llevarán cómodamente con ellos. Se trata de un dispositivo de manos libres, con alcance suficiente desde cualquier punto de la casa, con el que los usuarios podrán dormir sin problemas e incluso bañarse.
- una Central de Atención permanente, es decir, siempre operativa. Esta central dará respuesta a la necesidad de que se trate por sí misma o movilizand los recursos humanos o materiales de emergencia de la comunidad (policía, sanitarios...). La Central tendrá datos de cada usuario con información actualizada sobre su dependencia o situación sanitaria.
- un terminal conectado en todo momento, a través de la luz y de la línea telefónica, con el Centro de Atención. Este equipo que se instala en casa es pequeño, cómodo y fácil de usar.

Cuando el usuario/a presiona el botón se produce una llamada telefónica que será atendida por el personal de la central de Atención. Se producirá una comunicación verbal dirigida a conocer la situación de la persona mayor.^[9]

2.3 Conclusiones del estudio de mercado

Tal y como hemos explicado, es un hecho que las personas necesitamos una atención y un cuidado especial a partir de cierta edad. La mayor parte de estas personas que necesitan una asistencia continua tienen alguna persona de confianza que está a su cargo, ya sea un hijo, hermano, nieto,... pero estas personas no pueden estar todo el día pendientes o a cargo de ellos. Por eso se necesita algún medio a través del cual puedan estar convencidos de que estas personas estarán seguras mientras ellos siguen con su día a día.

Actualmente, hay servicios que solventan en parte el problema que estamos planteando, pero no hay ninguno completo.

Vamos a analizar las dos principales empresas que podrían hacernos competencia:

TELEASISTENCIA

Es el medio más utilizado por las personas mayores de 65 años en España, alcanzando más de 1,5 millones de usuarios, cifra que sube significativamente año tras año.

Ofrece una buena solución que utilizan muchos usuarios, en algunas comunidades autónomas está financiada parcialmente por la seguridad social, y está en colaboración con la Cruz Roja; pero no es una solución completa, ya que no recoge datos automáticamente, necesita de una comunicación con el paciente para saber el problema que está causado y necesita de terceras personas, como sanitarios u otros servicios públicos, para actuar en caso de emergencia.

LIVELY

Esta empresa es la segunda que tenemos que observar bien de cerca.

Ofrece una solución muy parecida a la que nosotros proponemos, ya que utiliza sensores y alertas directas para comunicarse con el paciente en una primera instancia, y tiene una base de datos en la nube para que el paciente o alguien a su cargo pueda consultar los datos del paciente en cualquier momento que desee, ya que esos datos se actualizan en tiempo real.

Aun así, no es una solución completa, ya que los sensores que tiene pensados esta empresa son limitados a la salud corporal del paciente, no tiene en cuenta la seguridad para evitar problemas dentro del hogar del paciente. Los sensores que dispone no están adaptados para evitar riesgos o peligros, y nuestra solución no solo quiere ayudar al paciente a llevar mejor su salud, sino que va más allá, y busca evitar los peligros de un hogar a una persona mayor o vulnerable, para que pueda sentirse segura en su hogar sin necesidad de un cuidado constante por parte de otra persona.

Otro problema de esta solución es que aún no está implantada en España, y no hay estimada ninguna fecha para que llegue a nuestro país.

3. FORMULACIÓN DEL PROBLEMA

Cada vez hay más población que necesita una atención en España, y eso es debido a que la esperanza de vida es mayor (véase Fig.1).

Las personas cada vez viven más años, y por eso hay que preocuparse por todas esas personas que alcanzan una cierta edad. El hacerse mayor significa que disminuyen muchas capacidades, como la percepción de nuestro entorno, y por eso tenemos que adaptar el entorno a las nuevas capacidades de esas personas.

Años*	Total España	65 años y más		65-79 años		80 años y más	
	Absoluto	Absoluto	% respecto al total	Absoluto	% respecto al total	Absoluto	% respecto al total
1900	18.618.086	967.774	5,2%	852.389	4,6%	115.385	0,6%
1910	19.995.686	1.105.569	5,5%	972.954	4,9%	132.615	0,7%
1920	21.389.842	1.216.693	5,7%	1.073.679	5,0%	143.014	0,7%
1930	23.677.794	1.440.744	6,1%	1.263.632	5,3%	177.112	0,7%
1940	26.015.907	1.699.860	6,5%	1.475.702	5,7%	224.158	0,9%
1950	27.976.755	2.022.523	7,2%	1.750.045	6,3%	272.478	1,0%
1960	30.528.539	2.505.165	8,2%	2.136.190	7,0%	368.975	1,2%
1970	34.040.989	3.290.800	9,7%	2.767.061	8,1%	523.739	1,5%
1981	37.683.362	4.236.740	11,2%	3.511.599	9,3%	725.141	1,9%
1991	38.872.268	5.370.252	13,8%	4.222.384	10,9%	1.147.868	3,0%
2001	40.847.371	6.958.516	17,0%	5.378.194	13,2%	1.580.322	3,9%
2011	46.815.916	8.116.347	17,3%	5.659.442	12,1%	2.456.908	5,2%
2021	46.037.605	9.466.481	20,6%	6.462.726	14,0%	3.003.755	6,5%
2031	45.351.545	11.903.963	26,2%	8.044.599	17,7%	3.859.364	8,5%
2041	44.680.774	14.791.516	33,1%	9.531.604	21,3%	5.259.912	11,8%
2051	43.581.814	16.486.938	37,8%	9.327.682	21,4%	7.159.256	16,4%
2061	41.603.330	16.095.184	38,7%	7.326.273	17,6%	8.768.911	21,1%

* De 1900 a 2011 los datos son reales; de 2021 a 2061 se trata de proyecciones

Fuente: INE: INEBASE:

1900-2001: Cifras de población.

2011: Cifras de población. Resultados nacionales, Censos de Población y Viviendas 2011

2021-2061: Proyecciones de población. Consulta en noviembre 2014

Fig. 2 Evolución de la población mayor, 1900-2061

Hacerse mayor no tiene que significar que ya no puedas desenvolverte autónomamente en tu entorno, o que tengas que depender de otras personas para poder llevar a cabo una rutina en tu día a día.

Para poder permitir a todas estas personas mayores, o a personas vulnerables que necesitan de una atención especial, una alta calidad de vida, hay que controlar su entorno y adaptarlo a sus necesidades y características.

3.1 Objetivo principal

El objetivo principal de este proyecto es muy sencillo y a la vez muy importante: aumentar la calidad de vida de todas esas personas que necesiten de una atención especial.

Para aumentar la calidad de vida de una persona con estas características, lo principal es aumentar la seguridad y bajar al mínimo todos los riesgos que pueda tener un hogar, para que pueda vivir cómodamente sin necesidad de depender de una tercera persona para hacer determinadas acciones o comprobar diferentes peligros.

También habrá una persona responsable que querrá estar al cuidado de esta persona que necesita atención especial, y para ello le facilitaremos una plataforma donde podrá consultar los datos a tiempo real.

Ofreceremos una solución, que será implantar un conjunto de sensores y actuadores en el hogar de esos pacientes para que recojan a tiempo real datos de diferentes peligros en varios puntos diferentes de la casa, y se actué a consecuencia teniendo en cuenta el grado de peligro en cada caso.

3.2 Objetivos específicos

Con tal de cumplir el objetivo principal, deberemos cumplir unos objetivos específicos que definiremos a continuación:

- ❖ Detectar posibles peligros de la casa instalando sensores en los puntos débiles en seguridad de la casa, tales como la cocina, la puerta de entrada a la vivienda o la sala donde esté colocada la lavadora.
- ❖ Controlar de manera inmediata los problemas de seguridad que se puedan originar en el hogar.
- ❖ Mantener una base de datos actualizada para que una persona responsable pueda ver en tiempo real los datos en alguna plataforma web o móvil.
- ❖ Tener la posibilidad de alertar a un equipo de emergencias (policía, bomberos,...) en caso de una emergencia en el hogar del paciente, tal como una fuga de gas o un incendio.

4. ALCANCE

El sistema resultante de este proyecto será un conjunto de aplicaciones software y unos módulos hardware basados en *Arduino*.^[10]

Se instalarán unas pequeñas señales luminosas o acústicas en diferentes sitios dentro del hogar para poder avisar a las personas que estén en ese momento dentro de la vivienda del peligro que se ha detectado.

Si no se da alguna señal de que una persona física ha detectado el problema y lo está solucionando, se procederá a avisar a los servicios de actuación de emergencia.

Los módulos de *Arduino* se instalarán a partir de las necesidades del paciente, pero los principales serán:

❖ **MÓDULO DE DETECCIÓN DE FUEGO**

Éste módulo se instalará en la cocina.

Constará de un sensor de fuego conectado a una placa *Arduino UNO* y se instalará un actuador para abrir el paso del agua del aspersor de la cocina.

Problemas que soluciona:

- Quedarse dormido mientras se cocina
- Irse de casa mientras se cocina
- Olvidarse de estar cocinando y ponerse a hacer otras cosas

Como los soluciona:

- Encenderá las alarmas de la casa
- Encenderá los aspersores
- Si la alarma no se desactiva en un minuto o no cesa el fuego, se avisará a los servicios de emergencia

❖ **MÓDULO DE FUGA DE GAS**

Éste módulo también se instalará en la cocina.

Constará de un sensor de gas conectado a una placa *Arduino UNO* y se instalará un actuador para cerrar el paso del gas y otro para abrir la ventana.

Problema que soluciona:

- Dejarse el gas abierto

Como lo soluciona:

- Encenderá las alarmas de la casa
- Se abrirán ventanas y se cerrará el gas
- Si la alarma no se desactiva en un minuto o no cesa el fuego, se avisará a los servicios de emergencia

❖ **MÓDULO DE INUNDACIÓN**

Éste módulo se instalará en la habitación donde esté situada la lavadora, el lavavajillas y la secadora. Si están en diferentes habitaciones, se instalará en todas ellas.

Constará de un sensor de inundación conectado a una placa *Arduino UNO*.

Problema que soluciona:

- La persona que habita la casa puede caerse por el agua
- La inundación puede ser causada por una fuga en la lavadora, en el lavavajillas o en la secadora

Como lo soluciona:

- Encenderá las alarmas de la casa
- Si la alarma no se desactiva en un minuto o no cesa el fuego, se avisará a los servicios de emergencia

❖ **MÓDULO DE LA PUERTA PRINCIPAL**

Éste módulo se instalará en la puerta principal de entrada a la vivienda.

Constará de un sensor magnético conectado a una placa *Arduino UNO* y se instalará un actuador para cerrar la puerta.

Problema que soluciona:

- Irse de casa y olvidarse la puerta abierta
- Irse a dormir y olvidarse la puerta abierta
- Olvidarse la puerta demasiado tiempo abierta

Como lo soluciona:

- Se cerrará la puerta
- Se avisará a la persona responsable

Todos estos módulos estarán conectados a través de un sistema de comunicación inalámbrico a una *Raspberry Pi*, donde estará montado un servidor. Desde este servidor se actuará de la manera que corresponda, ya sea avisando a la persona responsable del paciente o a los servicios de emergencia según sea el caso. También será desde donde se controlarán los datos para hacerlos accesibles en tiempo real.

Habrán diferentes formas de avisar a la persona responsable, podrá elegir un método u otro a partir de sus propias preferencias. Aun así, siempre podrá consultar los datos y avisos en tiempo real entrando en el portal Web o en la aplicación móvil que le corresponda. Estas serán las opciones para visualizar los datos y avisos:

- ❖ Enviar un mensaje SMS a un teléfono móvil facilitado (solo para avisos)
- ❖ Información visible desde aplicación web
- ❖ Información visible desde aplicación móvil

4.1 Posibles obstáculos

Los posibles obstáculos que se han tenido en cuenta y a los que posiblemente tendremos que hacer frente se han agrupado de la siguiente manera:

Internet

El acceso a internet es bastante básico para nuestro sistema por dos razones: la primera es que es un sistema que actualiza datos en tiempo real y los sube a la nube, y la segunda es que la persona responsable de la persona que se está monitorizando tiene que tener acceso a datos que están en la nube. Si cualquiera de las dos partes no tiene internet, habrá que buscar un sistema alternativo para tratar los datos, o buscar la forma de instalar una red de internet. Aun así nos basamos en que la mayor parte de las viviendas Españolas cuentan con acceso a internet.

Tecnologías

Utilizaré dos tecnologías de las cuales solo tengo algún conocimiento básico: *Android* y *Arduino*. Esto también puede ser un problema, ya que al solo tener conocimientos básicos puedo encontrarme problemas que no sepa solucionar y puedo perder mucho tiempo buscando una solución a ese problema.

Aplicación multiplataforma

La aplicación web no será un problema en un principio, pero la aplicación móvil solo se podrá desarrollar para *Android* por falta de tiempo. En un futuro se seguirá con el desarrollo para iPhone, pero se ha descartado el desarrollo para los sistemas de *Windows Phone* y *Blackberry* por falta de volumen de usuarios.

Tiempo

El TFG (Trabajo Final de Grado) tiene un tiempo límite en el cual se tiene que presentar, y ese será el principal obstáculo que tendré, ya que cualquier otro obstáculo que me encuentre por el camino retrasará el progreso del trabajo y hará que tenga que replantearme el contenido y los tiempos del mismo.

5. METODOLOGÍA Y RIGOR

5.1 Métodos de trabajo

Se utilizará una metodología en cascada para el desarrollo del proyecto. Éste modelo se define como una secuencia de actividades a ser seguidas en orden, donde la estrategia principal es definir y seguir el progreso del desarrollo de software hacia puntos de revisión bien definidos, es decir, se codifica y reparan los errores; es un proceso continuo de codificación y reparación. Sus características principales son:

- Es lineal
- Las actividades están relacionadas secuencialmente
- Cada etapa tiene una entrada y una salida
- Es rígido y sistemático: La entrada de una actividad es la salida de la etapa anterior, por lo cual no se puede dar inicio a la siguiente fase.
- Es monolítico: Existe una única fecha de entrega.
- La implementación se pospone hasta que no se comprendan los objetivos.
- Los documentos a entregar rigen el proceso de software
- Las fases que contempla el modelo de la cascada son al Análisis y especificación de requerimientos, diseño, codificación, integración y pruebas, liberación y mantenimiento.

Su ciclo de vida abarca las siguientes actividades:

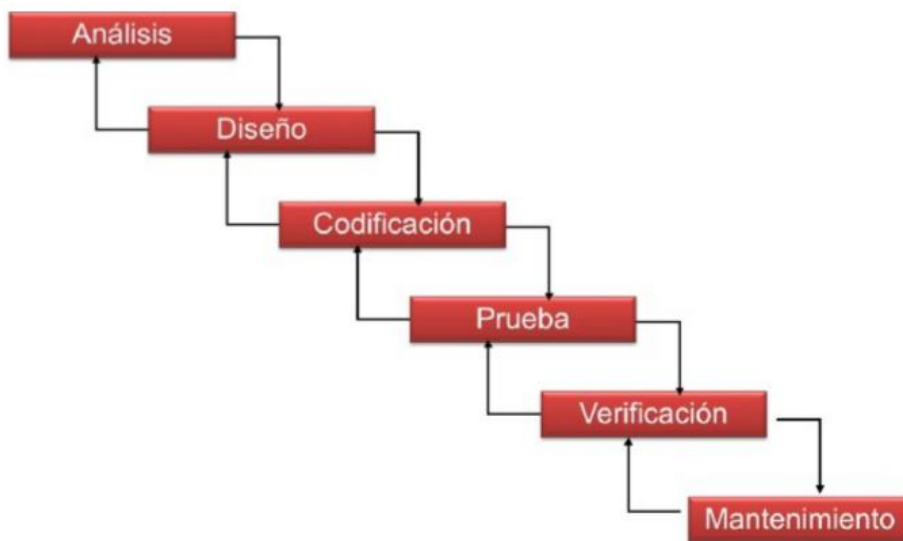


Fig. 3 Fases de la metodología en cascada

Se caracteriza por cumplir un orden secuencial en el desarrollo de sus tareas esto implica retardar el avance del proyecto ya que cada etapa inicia cuando haya finalizado la anterior siempre y cuando se haya realizado la evaluación respectiva y resuelto los errores en caso de que los hubiera tenido.

5.2 Herramientas de seguimiento

Este trabajo se desarrollará por una sola persona, así que no se han tenido en cuenta ninguna herramienta para trabajar simultáneamente con otras personas.

No obstante, se quiere tener un control de las versiones del proyecto para poder garantizar la recuperación de código por si ocurre algún fallo. Se utilizará la herramienta de gestión de repositorios ‘*BitBucket*’.^[11]

Para la comunicación con el ponente y el director del trabajo, se utilizará herramientas habituales como el correo electrónico.

Para la comunicación interna dentro de la empresa de trabajo, para posibles dudas desarrollando código o preguntas de conceptos, también se utilizará una herramienta de mensajería instantánea llamada ‘*Lync*’, de *Microsoft*.^[12]

5.3 Métodos de validación

Para validar el software desarrollado y todos los componentes que formarán este proyecto, se llevarán a cabo una serie de pruebas de *testing*, tanto por mi parte desarrollando las diferentes partes del proyecto, como del equipo interno de pruebas de la empresa.

Para validar los conceptos incluidos en el proyecto y el alcance del mismo, se llevarán a cabo reuniones regulares con el ponente y el director del proyecto.

Para garantizar la validez y resolver dudas sobre el desarrollo de este proyecto, se tendrá una puerta abierta para contactar con expertos en diferentes ámbitos dentro de la empresa.

6. PLANIFICACIÓN TEMPORAL

6.1. Planificación general

Planificación estimada del proyecto

El proyecto tiene una duración estimada de diez meses, con el inicio el día 1 de Septiembre y con fecha límite aproximada en Junio.

Dejamos un margen de un mes aproximado, para poder solucionar cualquier posible inconveniente que se encuentre en la realización del proyecto y para preparar la presentación final de éste.

Tiene una carga de trabajo de 735 horas aproximadamente, englobando todas las tareas necesarias para la realización del proyecto.

Recursos

Estos son los recursos que se van a utilizar en la realización del proyecto:

RECURSOS PERSONALES

Una persona, con una dedicación semanal de 20 horas durante todo el periodo de desarrollo del proyecto.

RECURSOS MATERIALES

Recursos	Tipo	Finalidad
Ordenador sobremesa con Windows 7 de 64 bits, procesador Intel i5-2400 a 3,1GHz y 8GB de memoria RAM	Herramienta de desarrollo	Para desarrollar las aplicaciones y la Memoria del proyecto.
Ordenador de sobremesa con Windows 8.1 de 64 bits, procesador Intel i5-4570 a 3,2 GHz y 8GB de memoria RAM	Herramienta de desarrollo	Para desarrollar las aplicaciones y la Memoria del proyecto.
One Plus TWO	Herramienta de desarrollo	Para testear aplicación móvil
Samsung Galaxy S4 MINI	Herramienta de desarrollo	Para testear aplicación móvil
Android estudio	Herramienta de desarrollo	Para desarrollar la aplicación Android
Sublime Text 2	Herramienta de desarrollo	Para desarrollar código
Eclipse	Herramienta de desarrollo	Para desarrollar código
Arduino	Herramienta de desarrollo	Para desarrollar código
Microsoft Office 2010	Herramienta de desarrollo	Para desarrollar la documentación del proyecto
Git	Herramienta de control	Para tener un control de las versiones del proyecto
Correo electrónico Gmail	Herramienta de comunicación	Para comunicación con el ponente y terceras personas
Microsoft Outlook	Herramienta de comunicación	Para comunicación con el director y otras personas internas de la empresa
Google Drive	Herramienta de desarrollo	Para desarrollar documentación y tener una copia de seguridad de la documentación del proyecto.
Gantter	Herramienta de gestión	Para planificar el proyecto. Se utiliza una adaptación para Google Drive.
Adobe Reader	Herramienta de desarrollo	Para la visualización de documentos en formato PDF
Postman	Herramienta de desarrollo	Para testear la API creada
Arduino UNO	Hardware	Para desarrollar los módulos de detección y los actuadores
Sensores de fuego, gas, agua, magnético	Hardware	Para desarrollar los módulos de detección
Hardware necesario para los actuadores	Hardware	Hardware no especificado necesario para desarrollar los actuadores
Raspberry Pi	Hardware	Para desarrollar el módulo donde estará instalado el servidor

Tabla 1. Recursos materiales

Plan de acción y valoración de alternativas

Tanto la planificación del proyecto como la fecha de entrega se prevén flexibles y modificables, dependerán del ritmo al que se avance desarrollando el proyecto y las diferentes aplicaciones.

Se tiene en cuenta un margen de 20 días de retraso en la fecha de entrega del proyecto, si con este cambio se consigue alguna mejora importante o alguna implementación extra.

La Memoria del proyecto se tendrá que efectuar como mínimo una semana antes de la defensa del proyecto, que está pensada para ser entre el 27 de Junio y el 1 de Julio según la web de la FIB. Cualquier cambio en esa planificación por parte de la universidad, implicará de manera directa un cambio en la planificación de este proyecto.

Se podrán producir cambios en la idea principal del proyecto. Estos cambios vendrán ocasionados por desviaciones temporales en alguna de las iteraciones del proyecto, y podrán ser por exceso o por defecto de tiempo. Se puede aumentar el tiempo estimado para alguna de las tareas, o bien reducirlo si se da por finalizada antes. Si en alguna de estas tareas, existe una desviación temporal de más de 3 días, se aplicará un plan de medidas específico para cada caso.

En caso de una desviación por exceso de tiempo, se podrá reducir el tamaño y dificultad de la misma iteración o de alguna posterior, o bien reducir la extensión del proyecto quitando alguna historia de usuario que no sea importante.

Si la desviación viene causada por defecto, entonces se podrá dar más tiempo a alguna otra iteración que lo necesite, volver a retomar alguna iteración anterior a la que se le haya quitado alguna extensión por falta de tiempo, o replantear alguna extensión o modificación.

Todas estas modificaciones del planteamiento se harán según cada caso específico. De esta manera aseguramos que se tratará cada caso según corresponda con su nivel de riesgo o importancia.

Consideraciones globales

Hace falta poner hincapié en que este proyecto será desarrollado por una sola persona, y no se podrá paralelizar ninguna tarea.

Además, debido a la naturaleza del proyecto y a la metodología utilizada (método en cascada), la elaboración de todas las iteraciones marcadas en el proyecto se tendrán que hacer de manera secuencial, exceptuando el montaje y el desarrollo de código de cada módulo de sensores y actuadores, que cada módulo será un pack que se tendrá que hacer conjuntamente ya que el código depende de cómo se monte. Debido a eso, cada tarea dependerá de la finalización de la tarea anterior, y se llevarán a cabo en el orden en que se han planificado.

Es por eso que, dado que el único camino que hay para desarrollar el proyecto es el camino crítico, i que el tiempo total del proyecto se corresponde a la suma de las horas de todas las tareas, es totalmente innecesario hacer un diagrama de *Pert*.

6.2 Descripción de las tareas

Búsqueda y puesta en marcha

En esta fase se hará una recogida y búsqueda de información. Por una parte, el jefe del proyecto informará del ámbito y alcance del proyecto; por otra, se buscará la información necesaria para desarrollar el proyecto.

Esta fase no tiene ninguna dependencia de precedencia.

Gestión del proyecto

Esta fase consta de la elaboración de los documentos de gestión del proyecto incluidos a la asignatura de GEP. Las tareas concretas del bloque y las duraciones estimadas de cada una se extraen de las indicaciones de la asignatura. Esta fase no tiene ninguna dependencia de precedencia, pero sí un calendario a seguir, estipulado por cada una de las entregas que marcan los profesores de la asignatura, y son fechas que no se pueden alargar o cambiar.

Creación de los módulos Arduino

Las dependencias de precedencia de esta etapa son las anteriores.

En este bloque de desarrollo se basa en desarrollar todos los módulos pensados para llevar a cabo el proyecto.

Para cada módulo habrá que:

1. Montar un módulo con una placa *Arduino UNO* con su sensor y actuador específico.
2. Habrá que desarrollar el código para que la placa de *Arduino* controle y actúe consecuentemente con los datos recogidos.

Estos son los diferentes módulos que hay que desarrollar:

- Módulo de detección de fuego
- Módulo de fuga de gas
- Módulo de inundación
- Módulo de la puerta principal
-

Desarrollo del servidor

Las dependencias de precedencia de esta etapa son las anteriores.

La componen tres fases: análisis de requisitos, especificación y diseño, implementación y pruebas. Cada una de estas fases tiene como dependencia de precedencia la anterior, y se van realizando de manera secuencial.

Una vez finalizada la aplicación, se valorará si hace falta añadir o mejorar alguna funcionalidad.

Desarrollo de la aplicación web

Las dependencias de precedencia de esta etapa son las anteriores.

La componen tres fases: análisis de requisitos, especificación y diseño, implementación y pruebas. Cada una de estas fases tiene como dependencia de precedencia la anterior, y se van realizando de manera secuencial.

Una vez finalizada la aplicación, se valorará si hace falta añadir o mejorar alguna funcionalidad.

Desarrollo de la aplicación móvil

Las dependencias de precedencia de esta etapa son las anteriores.

La componen tres fases: análisis de requisitos, especificación y diseño, implementación y pruebas. Cada una de estas fases tiene como dependencia de precedencia la anterior, y se van realizando de manera secuencial.

En cada una de las fases se tendrá en cuenta todos los cambios que se puedan generar en la aplicación web debido a que afectará a la aplicación móvil.

Una vez finalizada la aplicación, se valorará si hace falta añadir o mejorar alguna funcionalidad.

Mejoras

Opcional.

La dependencia de precedencia de esta etapa son las anteriores.

Esta fase depende del tiempo que se haya invertido en las fases anteriores. Si queda tiempo disponible para poder añadir alguna mejora a alguna funcionalidad o añadir alguna funcionalidad extra, se hará en esta iteración.

El análisis y la planificación detallada de esta parte se efectuará cuando se pueda comprobar de cuánto tiempo se dispone.

Documentación y presentación

En esta fase se terminará la documentación, que se irá completando a lo largo de las anteriores fases, y se preparará la defensa del proyecto delante del tribunal. La defensa tendrá lugar en una fecha posterior a la entrega de este documento.

La Memoria del proyecto incluirá la documentación del módulo de GEP, revisada y corregida según los criterios y las correcciones del profesor de esta asignatura, y toda la documentación derivada de las diferentes partes de cada fase del proyecto.

El cierre del proyecto, la finalización de la Memoria y la presentación final tienen como precedencia el resto de tareas del proyecto.

6.3 Calendario

Estimación de horas

Tarea	Horas
Búsqueda y puesta en marcha	25
Gestión del proyecto	75
Definición del alcance y contextualización	21
Planificación temporal	8,25
Gestión económica y sostenible	9,25
Presentación preliminar	6,25
Pliego de condiciones	12
Presentación oral y documento final	18,25
Creación de los módulos <i>Arduino</i>	250
Análisis de requisitos	50
Diseño y Especificación	50
Implementación y Pruebas	150
Desarrollo del servidor	80
Análisis de requisitos	15
Diseño y Especificación	25
Implementación y Pruebas	40
Desarrollo de la aplicación web	125
Análisis de requisitos	25
Diseño y Especificación	37,5
Implementación y Pruebas	62,5
Desarrollo de la aplicación móvil	125
Análisis de requisitos	25
Diseño y Especificación	37,5
Implementación y Pruebas	62,5
Mejoras	-
Análisis y Diseño	-
Implementación y pruebas	-
Documentación y presentación	55
Redacción de la Memoria	40
Presentación oral	15
Total	735

Tabla 2. Estimación de horas

Diagrama de Gantt

Incluido en el Anexo 1.

7. GESTIÓN ECONÓMICA

Ya tenemos planteado el problema y hemos diseñado la solución, y ahora es necesario hacer un estudio económico y un estudio del impacto social y ambiental, para determinar si el proyecto es viable o no.

7.1. Consideraciones iniciales

En este apartado se hace un análisis de los recursos necesarios para la realización del proyecto y una estimación de los costes de éste. Se harán estos costes en dos casos:

Caso 1: Teniendo en cuenta que soy becario, éste caso se hará con costes reales de recursos humanos.

Caso 2: Ésta segunda variante se hará teniendo en cuenta un salario real aproximado de cada rol.

7.2. Identificación y estimación de costes

La identificación y estimación de costes es el primer paso para la elaboración de un presupuesto económico. El primer recurso que tenemos que identificar y estimar es el recurso humano.

La remuneración de cada rol corresponde con el precio medio por hora que se establece en el informe de *Page Personnel* del año actual 2016. Suponemos que se trabaja 251 días al año. ^[13]

Suponemos que cada rol desarrolla las siguientes etapas:

- Jefe de proyecto: Búsqueda y puesta en marcha, Gestión del proyecto y Documentación y presentación.
- Analista: Todos los análisis de requisitos.
- Diseñador: Todos los diseños y especificaciones.
- Programador: el 60% de todas las implementaciones y pruebas.
- Responsable de pruebas: el 40% de todas las implementaciones y pruebas.

Rol	Horas estimadas	Remuneración (€/h) (Caso 1)	Remuneración (€/h) (Caso 2)	Coste estimado (€) (Caso 1)	Coste estimado (€) (Caso 2)
Jefe de proyecto	155	7,5	19,92	1.162,5	3.087,6
Analista	115	7,5	17,43	862,5	2.004,45
Diseñador	150	7,5	13,45	1.125	2.017,5
Programador	189	7,5	13,45	1.417,5	2.542,05
Responsable de pruebas	126	7,5	12,95	945	1.631,7
Total estimado	735			5.512,5	11.283,3

Tabla 3. Costes recursos humanos

Costes directos por actividad

A continuación, se puede ver con más detalle el coste que se puntuaba en el punto anterior, desglosado por cada actividad del proyecto.

Como en el punto anterior hemos indicado quien llevará a cabo cada tarea, no volveremos a detallarlo en este apartado.

Tarea	Horas estimadas	Coste estimado (€) (Caso 1)	Coste estimado (€) (Caso 2)
Búsqueda y puesta en marcha	25	187,5	498
Gestión del proyecto	75	562,5	1.494
Definición del alcance y contextualización	21	157,5	418,32
Planificación temporal	8,25	61,88	164,34
Gestión económica y sostenible	9,25	69,38	184,26
Presentación preliminar	6,25	46,88	124,5
Pliego de condiciones	12	90	239,04
Presentación oral y documento final	18,25	136,88	364,14
Creación de los módulos Arduino	250	1.875	3.531,5
Análisis de requisitos	50	375	871,5
Diseño y Especificación	50	375	672,5
Implementación y Pruebas	150	1.125	1.987,5
Desarrollo del servidor	80	600	1.127,7
Análisis de requisitos	15	112,5	261,45
Diseño y Especificación	25	187,5	336,25
Implementación y Pruebas	40	300	530
Desarrollo de la aplicación web	125	937,5	1.768,25
Análisis de requisitos	25	187,5	435,75
Diseño y Especificación	37,5	281,25	504,38
Implementación y Pruebas	62,5	468,75	828,12
Desarrollo de la aplicación móvil	125	937,5	1.768,25
Análisis de requisitos	25	187,5	435,75
Diseño y Especificación	37,5	281,25	504,38
Implementación y Pruebas	62,5	468,75	828,12
Mejoras	-	-	-
Análisis y Diseño	-	-	-
Implementación y pruebas	-	-	-
Documentación y presentación	55	412,5	1.095,6
Redacción de la Memoria	40	300	796,8
Presentación oral	15	112,5	298,8
Total	735	5.512,5	11.283,3

Tabla 4. Costes directos por actividad

Costes indirectos

Hay algunos gastos indirectos utilizados como la luz, el internet, el local,... no se tendrán en cuenta, ya que el trabajo se realiza en la empresa donde estoy trabajando de becario, y no se calcularán porque serían irrelevantes y además también sería un trabajo demasiado costoso calcular estos coste, ya que es una empresa grande y habría que tener en cuenta el recibo de luz por las personas que trabajan en el edificio y el tiempo que pasan en él.

Otros gastos indirectos, en cambio, si hay que tenerlos en cuenta:

- **Transporte:** se utilizará un coche (Ford Fiesta, 2004, 68cv) para ir de *Caldes de Montbui*, localidad del autor del proyecto, a *Barcelona*, ciudad donde está situada la empresa; y el metro de Barcelona (TMB) para ir del parking (gratuito), situado cerca de *Torras i Bages*, a *Glòries*, localización de la empresa. La distancia es de 25km aproximadamente, haciendo un total de 1000km mensuales, y calculando con una media entre Septiembre y Enero de 1,02558€/L ^[14], y un consumo de 4,4 l/100km ^[15], da un total de 45.13€ al mes. La tarjeta de metro, una T-Jove de 1 zona, se usará en su 80% para este proyecto, con un coste de 105€ cada tres meses ^[16].
- **Amortización del hardware:** Se utilizarán dos ordenadores de sobremesa, los dos tienen más de 4 años y su valor ya está amortizado. De los dos móviles que se usarán, el *Samsung Galaxy S4 Mini* ya está amortizado también, y el *One Plus Two* tiene un valor de 399,00€ ^[17], pero es mi móvil personal y solo contabilizaremos un 5% de su valor a este proyecto.
- **Impresiones en papel:** La entrega del proyecto incluye la entrega en papel de la documentación de este a cada uno de los miembros del tribunal. Supondremos una extensión aproximada de 200 páginas de Memoria para los 3 miembros del tribunal y el ponente del proyecto, a un coste de 0,05€ por página (incluyendo encuadernaciones).
- **Software:** No habrá ningún coste de Software, ya que una parte del Software utilizado era parte del Software ofrecido por la empresa o la universidad, y la otra parte era Software gratis.
- **Módulos de Arduino:** Material necesario para la implantación de los módulos de *Arduino*, se detalla a continuación, y el total sube a
 - Arduino UNO (x4) → 8.51€ c/u
 - Válvula agua y gas (x2) → 15€ c/u
 - Sensor de llama → 2€
 - Sensor de gas → 4€
 - Sensor del nivel del agua → 4,5€
 - Sensor magnético de alarma para puerta o ventana → 4,43€
 - Servo → 8,95€
 - Raspberry PI → 25,95 €
 - Zumbador → 2€
 - Leds → 1€
 - Wifi shield → 10,45€
 - Xbee shield (x4) → 6,98€

Producto	Unidades	Precio unitario	Porcentaje de dedicación	Coste estimado
Luz, internet, local...	7 meses	0€/mes	-	0 €
Transporte (Coche)	7 meses	45,13€/mes	100%	315,91 €
Transporte (Metro)	2,33 bonos (7 meses)	35€/mes	100%	245 €
Hardware (One Plus)	1 unidad	399€/unidad	5%	19,95 €
Hardware (el resto)	7 meses	0€/mes	-	0 €
Impresiones a Papel	800 páginas	0,05€/página	100%	40 €
Software	7 meses	0€/mes	-	0 €
Módulos de Arduino	19 piezas	-	-	155,24 €
Total				776,1 €

Tabla 5. Costes indirectos

Contingencia

Reservamos una parte del presupuesto para la partida de contingencia, concretamente un 15% de la suma de los costes directos e indirectos.

Presentación	Porcentaje	Precio (Caso 1)	Coste (Caso 1)	Precio (Caso 2)	Coste (Caso 2)
Costes directos	15%	5.512,5	826,88	11.283,3	1.692,5
Costes indirectos	15%	776,1	116,42	776,1	116,42
Total			943,3 €		1808,92 €

Tabla 6. Costes de contingencia

Imprevistos

Retraso de 20 Días: en el plan de acción hemos mencionado la posibilidad de un margen de 20 días en la fecha de entrega del proyecto. Suponiendo que el retraso consta de una implicación en horas a partes iguales de analistas y programadores, para calcular el precio del Caso 1 lo contabilizamos a 15€/hora (7,5€/hora el analista y 7,5€/hora el programador) y el precio del Caso 2 lo contabilizamos a 30,88€/hora (17,43€/hora del analista y 13,45€/hora del programador). Recordamos que tendremos una carga diaria de 4 horas. La probabilidad de que pase este imprevisto es de un 20%.

Avería de algún ordenador: en caso de que haya algún problema con el *hardware* de alguno de los ordenadores utilizados, se tendrá que reparar o substituir por otro, con un aumento en el coste del proyecto igual al precio del nuevo ordenador. Supondremos que el nuevo ordenador tendrá unas características parecidas al ordenador de la empresa. Este imprevisto tiene una baja probabilidad del 5%.

Avería del móvil: en caso de que haya un problema con el *Samsung*, habrá que reponerlo por otro, pero no tendrá coste alguno ya que en el departamento de pruebas de la empresa tienen más ejemplares de ese teléfono. En caso de que haya un problema con el *One Plus*, tampoco habrá que reponerlo porque al ser nuevo aún está en garantía. Estos imprevistos también tienen una probabilidad del 5%, pero no tendremos en cuenta estos imprevistos porque no supondrán coste alguno.

Imprevisto	Probabilidad	Unidades	Precio (Caso 1)	Coste (Caso 1)	Precio (Caso 2)	Coste (Caso 2)
Retraso de 20 días	20%	80 horas	15€/hora	240 €	30,88€/hora	494,08€
Avería de algún ordenador	5%	1	400 €	20 €	400 €	20 €
Total				260 €		514,08 €

Tabla 7. Costes de imprevistos

Presupuesto

Consideraciones finales:

- No se tienen en cuenta aumentos en el precio durante el proyecto, ya que no es de larga duración y no se prevé que se dé la situación
- No se añade al presupuesto ningún margen de beneficios sobre el coste total del proyecto, ya que es un proyecto sin ánimo de lucro y no destinado a la venta de ningún producto.

Concepto	Coste (Caso 1)	Coste (Caso 2)
Costes directos	5.512,5 €	11.283,3 €
Costes indirectos	776,1 €	776,1 €
Contingencia	943,3 €	1808,92 €
Imprevistos	260 €	514,08 €
Total	7491,9 €	14.382,4 €

Tabla 8. Presupuesto

7.3. Control de gestión

Como la parte de recursos no humanos es una inversión inicial, no se puede hacer un control constante ya que sólo se podrá comparar el coste final con el coste calculado en el presupuesto una vez acabado el proyecto. Aun así, puede haber una desviación respecto al coste del presupuesto inicial si se ha producido algún error a la hora de diseñar los módulos de *Arduino* o alguna otra etapa del proyecto.

La parte de los recursos humanos sí que se puede controlar, y se tiene que hacer un control constante. Para ello, se hará un análisis de los costes al final de cada etapa del proyecto, teniendo en cuenta las horas totales dedicadas y comparándolas con el presupuesto estimado en este documento.

En cualquiera de los dos casos, si se produce una diferencia notable de los costes respecto a los estimados, ya sea por pasarse de largo o por quedarse corto, se realizará un estudio para estudiar por qué se ha producido esa diferencia y se intentará corregir las estimaciones de presupuesto de las siguientes fases, si se cree que se ha producido un error similar, para crear un presupuesto más exacto.

Si en la suma de las diferencias de todas las fases sale como resultado que el presupuesto inicial se ha quedado corto, se tendrá que asignar el fondo calculado de contingencia para cubrir los gastos ocasionados.

Para calcular dichas desviaciones se hará de la siguiente manera:

- Desvío de mano de obra en precio = $(\text{coste estimado} - \text{coste real}) * \text{consumo de horas real}$
- Desvío en la realización de una tarea en precio = $(\text{coste estimado} - \text{coste real}) * \text{consumo horas reales}$
- Desvío de un recurso en precio = $(\text{coste estimado} - \text{coste real}) * \text{consumo real}$
- Desvío en la realización de una tarea en consumo = $(\text{consumo estimado} - \text{consumo real}) * \text{coste real}$
- Desvío total en la realización de tareas = $\text{coste total estimado tarea} - \text{coste total real tarea}$
- Desvío total en recursos = $\text{coste total estimado recursos} - \text{coste total real recursos}$
- Desvío total costes fijos = $\text{coste total coste fijos presupuestado} - \text{coste total fijo real}$

8. SOSTENIBILIDAD Y COMPROMISO SOCIAL

A continuación, presentamos el estudio sobre sostenibilidad que se ha hecho referente al proyecto. En la tabla presentamos los resultados, que se explican más adelante, en forma de la matriz propuesta en la guía de la asignatura (sólo la parte de planificación).

Sostenibilidad	Económica	Social	Ambiental	Total
Planificación	Viabilidad económica	Mejora de la calidad de vida	Análisis de recursos	
Valoración	8	9	6	23

Tabla 9. Valoración de sostenibilidad

8.1. Económica

Se ha realizado una evaluación de los costes del proyecto, tanto de recursos materiales como de recursos humanos, teniendo en cuenta el coste de reparaciones del material y de desviaciones del calendario dentro de la partida de imprevistos.

Su coste es realmente bajo a causa de su naturaleza académica, y forma parte de otro proyecto aún más grande dentro de la empresa en la que se realiza este proyecto. Aun así, su valor real dentro del mercado no es nada elevado, ya que se podría comercializar con un amplio mercado que cada día está en aumento, a un precio bajo y dentro de una plataforma más grande de lo que abarca este proyecto.

En cuestión de sostenibilidad económica, este proyecto merece una valoración de 8.

8.2. Social

El objetivo de este trabajo es ayudar y facilitar la vida a personas mayores o con alguna necesidad de ayuda social.

El proyecto está orientado a ayudar socialmente a esas personas que no se pueden valer de por sí mismas para garantizar su seguridad dentro de un hogar, y para facilitar a las familias de todas esas personas a tener la seguridad que sus familiares cercanos con necesidad de atención estarán seguros en su hogar.

Proveemos diferentes medios para que una persona que esté a cargo de otra persona que necesita una atención especial pueda consultar el estado de las viviendas de éstas, para que así pueda asegurarse de que no hay ningún peligro en su vivienda y poder saber a distancia y sin tener que estar en contacto con el paciente que éste está perfectamente.

También le damos al paciente esa autonomía que tanto espera tener una persona de esas condiciones, para que se sienta autosuficiente y no crea que tiene que ser una persona dependiente de otra.

Por todo eso, creemos que este proyecto merece una valoración de sostenibilidad social de un 9.

8.3. Ambiental

Para la parte de sostenibilidad ambiental, valoramos este proyecto con un 6, ya que prácticamente no aumentará el empuje ecológico que se quieren tener todos los proyectos actuales.

Se utilizará el transporte público en la medida de lo posible. No se prevé desmontar ningún recurso material en acabar el proyecto, ni se necesitarán materias primas ni ningún producto manufacturado nuevo.

El proyecto puede ayudar a la sostenibilidad ambiental reduciendo algún caso de catástrofe, como un incendio o fuga de gas, evitando que se tengan que gastar otros recursos materiales reparando los desperfectos que puedan contaminar y evitando también la contaminación que causarían los equipos de emergencia en su desplazamiento.

9. DESCRIPCIÓN GENERAL DE LOS REQUISITOS

En este apartado vamos a definir los requisitos del sistema a desarrollar, es decir, las condiciones que tiene que satisfacer el software que desarrollaremos. Se definen tanto los requisitos funcionales, como los requisitos de calidad.

9.1 Requisitos funcionales

En esta sección definiremos los requisitos funcionales del sistema, las funcionalidades que tienen que estar en el sistema a desarrollar. Para definirlos se seguirá la siguiente plantilla:

Campo	Especificación
Requisito #:	[Número del requisito funcional]
Descripción:	[Breve descripción del requisito]
Detalles adicionales:	[Descripción más detallada nombrando a los actores]
Partes interesadas:	[Partes interesadas que solicitan el objetivo asociado al requisito]
Prioridad:	[Indica si es obligatorio o es un requisito secundario]

Como partes interesadas, tendremos en cuenta tres partes:

- Empresa: la empresa que ofrece el sistema (Worldline).
- Paciente: persona de edad avanzada o con alguna discapacidad. El sistema estará instalado en casa de esta persona.
- Cliente: persona a cargo del paciente, normalmente será un familiar directo (hijos, hermanos, ...)

A continuación se muestran todos los requisitos funcionales:

Campo	Especificación
Requisito #:	1
Descripción:	Se tiene que detectar todas las fugas
Detalles adicionales:	El sistema debe ser capaz de detectar las diferentes fugas para poder actuar debidamente. Normalmente serán: fuego, gas, inundación y puerta abierta.
Partes interesadas:	Empresa, cliente, paciente
Prioridad:	Obligatorio

Campo	Especificación
Requisito #:	2
Descripción:	Se tienen que mostrar los datos de los sensores
Detalles adicionales:	El cliente querrá ver en todo momento el estado de la casa de la persona que tienen a su cargo, y habrá que mostrarles los datos que recojan los sensores en todo momento (plataforma web y móvil).
Partes interesadas:	Cliente
Prioridad:	Obligatorio

Campo	Especificación
Requisito #:	3
Descripción:	Se tendrá que activar una alarma
Detalles adicionales:	Si se detecta algún peligro en la casa, se tendrá que activar una alarma tanto acústica como lumínica para avisar al paciente que está ocurriendo algún peligro en la casa, y pueda poner solución o ponerse a salvo.
Partes interesadas:	Paciente
Prioridad:	Obligatorio

Campo	Especificación
Requisito #:	4
Descripción:	Se tendrá que actuar en caso de peligro
Detalles adicionales:	Si se detecta que hay algún peligro en la casa y, siguiendo el requisito 3, se activa la alarma, se tendrá que actuar según corresponda. Depende de que peligro se haya detectado, se actuará de una forma u otra.
Partes interesadas:	Paciente
Prioridad:	Obligatorio

Campo	Especificación
Requisito #:	5
Descripción:	Se tendrá que enviar notificaciones
Detalles adicionales:	En caso de que haya un peligro en la casa, se tendrá que enviar una notificación al teléfono móvil del cliente en caso de que este tenga la aplicación descargada en su teléfono móvil. El teléfono tendrá que ser un dispositivo <i>Smartphone</i> .
Partes interesadas:	Cliente
Prioridad:	Obligatorio

Campo	Especificación
Requisito #:	6
Descripción:	Se tendrá que avisar a emergencias
Detalles adicionales:	Sólo en caso de que el peligro que se detecte se prolongue durante más de un cierto tiempo, o si el peligro pasa a ser grave, se avisará a los servicios de emergencia para que se trasladen al domicilio del paciente y actúen urgentemente.
Partes interesadas:	Cliente, Paciente
Prioridad:	Obligatorio

Campo	Especificación
Requisito #:	7
Descripción:	El sistema tendrá que conectarse a otros sistemas
Detalles adicionales:	El sistema que ofreceremos tendrá que ser capaz de conectarse con otros sistemas mediante la nube.
Partes interesadas:	Empresa
Prioridad:	Obligatorio

Campo	Especificación
Requisito #:	8
Descripción:	Se deberá avisar de batería baja
Detalles adicionales:	Cada uno de los módulos <i>Arduino</i> estará conectado a una batería externa, y esta se puede acabar, así que para evitar que un módulo se quede inutilizado por quedarse sin batería, deberá avisar cuando su batería esté baja. Se procederá a avisar al Cliente para que reemplace esta batería, y si llega a un nivel bajo crítico, se avisará a la empresa para que se comunique con el cliente.
Partes interesadas:	Empresa, Cliente, Paciente
Prioridad:	Obligatorio

Campo	Especificación
Requisito #:	9
Descripción:	Se deberá avisar si se va la luz
Detalles adicionales:	El sistema estará conectado a un SAI (Sistema de Alimentación Ininterrumpida) para evitar que se quede sin comunicación con la nube o el exterior, y se avisará al Cliente que se ha ido la luz para que se pueda arreglar el problema y volver a funcionar con normalidad, ya que el SAI da una vida corta y si se acaba sin que vuelva la luz, el sistema se apagará y no podrá alertar de ningún peligro.
Partes interesadas:	Empresa, Cliente, Paciente
Prioridad:	Obligatorio

9.2 Requisitos no funcionales

Ahora vamos a explicar los requisitos no funcionales, que son propiedades o cualidades que el producto debe tener. Estos requisitos imponen unas restricciones en el diseño, en la implementación o en los estándares de calidad del producto.

En los requisitos no funcionales también vamos a utilizar un plantilla.

Prioridad:	[indica si el requisito es obligatorio o secundario]	Historias de usuario afectadas	[Enumeración de las historias de usuario que son afectadas por dicho requisito.]
Descripción:	[Breve descripción del requisito no funcional]		
Justificación:	[Explicación de la existencia de dicho requisito]		
Criterio de satisfacción:	[Criterio para indicar cómo o el por qué se satisface el requisito]		
Satisfacción del cliente:	[nivel de satisfacción del 1 al 5]	Insatisfacción del cliente:	[nivel de insatisfacción del 1 al 5]
Dependencias:	[si depende de otro requisito]	Conflictos:	[si entra en conflicto con otro requisito]
Partes interesadas:	[Partes interesadas que solicitan el objetivo asociado al requisito]		
Materiales de soporte:	[referencias para complementar el requisito]		
Historia:	[fecha de creación del requisito y de modificación si la hay]		

Una vez explicada la plantilla que se va a utilizar, se verá a continuación todos los requisitos no funcionales, agrupados por los siguientes grupos: percepción, rendimiento, funcionamiento, ambientales, seguridad, políticas y legales.

Requisitos de percepción (look and feel)

Apariencia

Requisito #:	10		
Prioridad:	Obligatorio	Historias de usuario afectadas	-
Descripción:	El diseño de la aplicación tiene que ser atractiva y sencilla, invitando a hacer uso de ella.		
Justificación:	Un buen diseño es indispensable para que cualquier persona pueda interactuar con la aplicación correctamente.		
Criterio de satisfacción:	Se siguen los patrones y normas establecidas por W3C.		
Satisfacción del cliente:	3	Insatisfacción del cliente:	5
Dependencias:	-	Conflictos:	-
Partes interesadas:	Cliente		
Materiales de soporte:	http://www.w3.org/TR/WAI-WEBCONTENT/		
Historia:	Creado el 10 de Marzo de 2016.		

Estilo

Requisito #:	11		
Prioridad:	Deseable	Historias de usuario afectadas	-
Descripción:	El sistema tiene que tener un diseño moderno y minimalista.		
Justificación:	La mayoría de las aplicaciones exitosas tienen un diseño moderno y minimalista que facilitan la interacción con el usuario.		
Criterio de satisfacción:	El 80% de los clientes afirman que se sienten más cómodos con interfaces limpias y simples.		
Satisfacción del cliente:	2	Insatisfacción del cliente:	3
Dependencias:	-	Conflictos:	-
Partes interesadas:	Cliente		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Requisitos de usabilidad y humanidad

Facilidad de uso

Requisito #:	12		
Prioridad:	Obligatorio	Historias de usuario afectadas	-
Descripción:	La aplicación móvil tiene que ser fácil de utilizar e intuitiva.		
Justificación:	La aplicación tiene que ser intuitiva y fácil de utilizar con tal de conseguir un sistema atractivo que mejore la experiencia del usuario.		
Criterio de satisfacción:	La aplicación solo incorpora las siguientes interacciones: pulsar y deslizar.		
Satisfacción del cliente:	3	Insatisfacción del cliente:	4
Dependencias:	-	Conflictos:	-
Partes interesadas:	Cliente		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Aprendizaje

Requisito #:	13		
Prioridad:	Obligatorio	Historias de usuario afectadas	-
Descripción:	La aplicación se tiene que poder utilizar sin formación previa.		
Justificación:	Se tiene que procurar que no haya una curva de aprendizaje para evitar que el usuario rechace utilizar la aplicación debido a su dificultad.		
Criterio de satisfacción:	Usuarios sin conocimientos en <i>Smartphones</i> saben utilizar la aplicación móvil.		
Satisfacción del cliente:	2	Insatisfacción del cliente:	5
Dependencias:	-	Conflictos:	-
Partes interesadas:	Clientes		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Comprensión y cortesía

Requisito #:	14		
Prioridad:	Obligatorio	Historias de usuario afectadas	-
Descripción:	La parte lingüística de la aplicación tiene que ser impecable, sin ambigüedades. Los textos redactados los tiene que comprender todos los usuarios.		
Justificación:	La aplicación la usarán personas de una gran variedad de edades, tenemos que redactar los textos de manera que se adapten a todos los usuarios y que no tengan ninguna dificultad en entenderlos.		
Criterio de satisfacción:	Auto-explicativo.		
Satisfacción del cliente:	2	Insatisfacción del cliente:	5
Dependencias:	-	Conflictos:	-
Partes interesadas:	Clientes.		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Requisitos de rendimiento

Latencia y velocidad

Requisito #:	15		
Prioridad:	Deseable	Historias de usuario afectadas	-
Descripción:	El tiempo máximo de respuesta de la aplicación tiene que ser de un segundo en el 90% de las operaciones.		
Justificación:	Un tiempo de respuesta rápido permite que el usuario no pierda el hilo de lo que está haciendo con el sistema.		
Criterio de satisfacción:	Auto-explicativo.		
Satisfacción del cliente:	3	Insatisfacción del cliente:	3
Dependencias:	-	Conflictos:	-
Partes interesadas:	Clientes.		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Requisito #:	16		
Prioridad:	Deseable	Historias de usuario afectadas	-
Descripción:	El sistema tiene que ser capaz de dar respuesta a una petición en menos de 2 segundos.		
Justificación:	Si el tiempo de respuesta es más lento, el usuario tiene la sensación de que la aplicación no funciona correctamente o que está perdiendo el tiempo.		
Criterio de satisfacción:	Auto-explicativo.		
Satisfacción del cliente:	3	Insatisfacción del cliente:	4
Dependencias:	-	Conflictos:	-
Partes interesadas:	Clientes.		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Fiabilidad y disponibilidad

Requisito #:	17		
Prioridad:	Obligatorio	Historias de usuario afectadas	Todos los que manipulen datos e información sensible
Descripción:	Los clientes tienen que poder acceder a la aplicación las 24 horas del día durante todo el año.		
Justificación:	El cliente tiene que poder acceder a la información de la vivienda del paciente en todo momento.		
Criterio de satisfacción:	El sistema estará disponible todo tiempo.		
Satisfacción del cliente:	3	Insatisfacción del cliente:	5
Dependencias:	-	Conflictos:	-
Partes interesadas:	Cliente y paciente.		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Capacidad

Requisito #:	18		
Prioridad:	Deseable	Historias de usuario afectadas	Todos los que manipulen datos e información sensible
Descripción:	El sistema tiene que soportar la conexión simultánea de hasta 500 usuarios.		
Justificación:	Se tiene que asegurar que el sistema es capaz de soportar un cierto volumen de clientes haciendo servir la aplicación a la vez.		
Criterio de satisfacción:	Auto-explicativo.		
Satisfacción del cliente:	1	Insatisfacción del cliente:	5
Dependencias:	-	Conflictos:	-
Partes interesadas:	Cliente y paciente.		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Escalabilidad o extensibilidad

Requisito #:	19		
Prioridad:	Deseable	Historias de usuario afectadas	Todos los que manipulen datos
Descripción:	La aplicación tiene que ser escalable y extensible.		
Justificación:	Tiene que poder añadir tantas entradas en la base de datos como se desee, así como nuevas funcionalidades a la aplicación o nuevos sensores o salas donde se encuentran estos sensores.		
Criterio de satisfacción:	Se puede añadir contenido correctamente y se pueden ampliar las funcionalidades cuando el cliente o la empresa lo deseen.		
Satisfacción del cliente:	3	Insatisfacción del cliente:	2
Dependencias:	-	Conflictos:	-
Partes interesadas:	Cliente, paciente y empresa.		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Requisitos de funcionamiento y ambientales

Entorno físico esperado

Requisito #:	20		
Prioridad:	Obligatorio	Historias de usuario afectadas	-
Descripción:	La aplicación se tiene que poder utilizar en todo tipo de espacio donde haya una conexión a internet.		
Justificación:	La aplicación de tiene que poder consultar en todo momento para asegurar que el cliente puede consultar los datos sensibles y la seguridad del paciente en todo momento que cumpla el requisito de tener un punto de acceso a internet.		
Criterio de satisfacción:	Auto-explicativo.		
Satisfacción del cliente:	3	Insatisfacción del cliente:	5
Dependencias:	-	Conflictos:	-
Partes interesadas:	Cliente, paciente y empresa.		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Entorno propio

Requisito #:	21		
Prioridad:	Deseable	Historias de usuario afectadas	-
Descripción:	La aplicación tiene que disponer de una aplicación web y de una aplicación móvil (tanto Android como iOS).		
Justificación:	Los clientes querrán consultar los datos tanto en una página web como desde un teléfono móvil. La mayoría de teléfonos móviles funcionan a través de estos dos sistemas operativos (Android y iOS), por lo tanto, la aplicación tiene que estar disponible para estos dos sistemas como mínimo.		
Criterio de satisfacción:	Existe una versión para Android y una para iOS.		
Satisfacción del cliente:	1	Insatisfacción del cliente:	5
Dependencias:	-	Conflictos:	-
Partes interesadas:	Cliente.		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Interfaz con sistemas adyacentes

Requisito #:	22		
Prioridad:	Obligatorio	Historias de usuario afectadas	Todos en los que se comunican dos o más sistemas diferentes
Descripción:	La aplicación tiene que ser capaz de interactuar con otros sistemas.		
Justificación:	Los diferentes sistemas implementados (Arduino, Servidor, aplicación Web, aplicación Móvil) tienen que ser capaces de interactuar entre ellos para poder tener la información de los datos actualizada al momento y de forma automática.		
Criterio de satisfacción:	Es posible ver la información actualizada en todo momento.		
Satisfacción del cliente:	4	Insatisfacción del cliente:	5
Dependencias:	-	Conflictos:	-
Partes interesadas:	Cliente, paciente, empresa.		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Requisito #:	23		
Prioridad:	Obligatorio	Historias de usuario afectadas	-
Descripción:	El sistema tiene que tener un periodo de pruebas antes de su lanzamiento.		
Justificación:	No se puede lanzar al mercado este sistema sin haberlo probado en su totalidad previamente y haber asegurado que no tiene errores.		
Criterio de satisfacción:	Probaremos todas las historias de usuario en un entorno preparado de pruebas, y luego lo trasladaremos para probar en un entorno real con gente real.		
Satisfacción del cliente:	3	Insatisfacción del cliente:	4
Dependencias:	-	Conflictos:	-
Partes interesadas:	Cliente, paciente, empresa.		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Requisitos de mantenimiento y soporte

Adaptabilidad

Requisito #:	24		
Prioridad:	Obligatorio	Historias de usuario afectadas	-
Descripción:	La aplicación tiene que ser compatible con versiones de Android 4.1.x (Jelly Bean) y posteriores (95,7% del mercado Android) y iOS 8.0 y posteriores (94% del mercado iOS). [Datos en el Anexo 2].		
Justificación:	Hemos escogido estas versiones para garantizar que llega al mayor número de clientes, debido a la diversidad de dispositivos móviles que hay en el mercado.		
Criterio de satisfacción:	Auto-explicativo.		
Satisfacción del cliente:	3	Insatisfacción del cliente:	5
Dependencias:	-	Conflictos:	-
Partes interesadas:	Cliente y empresa.		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Requisitos de seguridad

Acceso

Requisito #:	25		
Prioridad:	Obligatorio	Historias de usuario afectadas	-
Descripción:	El cliente tendrá acceso para ver los datos actualizados de los dispositivos que tenga enlazados a su cuenta, y la empresa (desarrolladores) tendrá acceso al código de todos los dispositivos y aplicaciones.		
Justificación:	Con tal de proteger los datos sensibles, el cliente no podrá ver ningún dato no autorizado ni manipular ningún dato.		
Criterio de satisfacción:	Auto-explicativo.		
Satisfacción del cliente:	3	Insatisfacción del cliente:	5
Dependencias:	-	Conflictos:	-
Partes interesadas:	Cliente y empresa.		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Integridad

Requisito #:	26		
Prioridad:	Obligatorio	Historias de usuario afectadas	-
Descripción:	El sistema no permite introducir datos en formatos incorrectos.		
Justificación:	El sistema necesita mantener una consistencia en los datos de tal manera que no se pueda introducir datos incorrectos.		
Criterio de satisfacción:	El sistema dará un mensaje de incompatibilidad cuando se ha introducido un dato con formato incorrecto.		
Satisfacción del cliente:	2	Insatisfacción del cliente:	5
Dependencias:	-	Conflictos:	-
Partes interesadas:	Cliente.		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Privacidad

Requisito #:	27		
Prioridad:	Obligatorio	Historias de usuario afectadas	-
Descripción:	La aplicación debe de proteger la información privada del cliente y del paciente de acuerdo con las leyes de privacidad pertinente y la política de información de la organización.		
Justificación:	El sistema tiene que asegurar que se cumplen todas las leyes relacionadas con la privacidad de los datos de un individuo.		
Criterio de satisfacción:	El sistema se compromete a respetar la legislación vigente de protección y confidencialidad en el tratamiento de datos de carácter personal, de acuerdo con el que dispone la Ley orgánica 15/1999, de 13 de Diciembre, de protección de datos de carácter personal, y su normativa de desarrollo.		
Satisfacción del cliente:	2	Insatisfacción del cliente:	5
Dependencias:	-	Conflictos:	-
Partes interesadas:	Cliente, paciente y empresa.		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Requisitos culturales y políticos

Culturales

Requisito #:	28		
Prioridad:	Obligatorio	Historias de usuario afectadas	-
Descripción:	Ninguna parte del sistema incluirá material ofensivo a religiones ni creencias de otras culturas.		
Justificación:	Ya que el sistema lo podrán utilizar clientes y pacientes con diversas creencias, tenemos que procurar que el contenido sea imparcial a cualquier tipo de creencia.		
Criterio de satisfacción:	No se recibirá ninguna queja de sentido cultural y/o religioso por parte de los clientes o de los pacientes.		
Satisfacción del cliente:	1	Insatisfacción del cliente:	5
Dependencias:	-	Conflictos:	-
Partes interesadas:	Clientes y pacientes.		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

Requisitos legales

Cumplimiento legal

Requisito #:	29		
Prioridad:	Obligatorio	Historias de usuario afectadas	-
Descripción:	La empresa tiene que tener los derechos de todo el material que se publique en la aplicación.		
Justificación:	Con tal de evitar conflictos con la ley, se tendrá que pactar los derechos con los propietarios de todos los contenidos que no sean propiedad de la empresa.		
Criterio de satisfacción:	Auto-explicativo.		
Satisfacción del cliente:	1	Insatisfacción del cliente:	2
Dependencias:	-	Conflictos:	-
Partes interesadas:	Empresa.		
Materiales de soporte:	-		
Historia:	Creado el 10 de Marzo de 2016.		

9.3 Requisitos de datos

Por último, vamos a explicar los requisitos de datos. Para explicar estos requisitos, volveremos a utilizar la misma plantilla que se utilizaba para los requisitos funcionales.

Los requisitos de datos se detallan a continuación:

Campo	Especificación
Requisito #:	30
Descripción:	El sistema debe obtener los datos de usuario de otro sistema.
Detalles adicionales:	El sistema debe obtener mediante el identificador de usuario el usuario en sí haciendo una petición a un sistema ya existente.
Partes interesadas:	Empresa.
Prioridad:	Obligatorio.

Campo	Especificación
Requisito #:	31
Descripción:	El sistema almacena la siguiente información de cada situación de riesgo: identificador, modulo, sensor, fecha, hora, alarma lanzada, alarma parada y lugar.
Detalles adicionales:	El identificador es único y auto-incremental, se utiliza a nivel interno. Los otros datos se obtienen a partir de los diferentes sistemas involucrados.
Partes interesadas:	Cliente, empresa.
Prioridad:	Obligatorio.

Campo	Especificación
Requisito #:	32
Descripción:	El sistema almacena la siguiente información sobre los usuarios de la web: nombre de usuario, contraseña.
Detalles adicionales:	Los usuarios de la web serán los clientes que tengan contratado nuestro sistema para un paciente a su cargo. El nombre de usuario debe de ser único.
Partes interesadas:	Cliente, empresa.
Prioridad:	Obligatorio.

10. ESPECIFICACIÓN

La especificación de requisitos describe el comportamiento esperado en el software una vez desarrollado. Gran parte del éxito de un proyecto de software radicaré en la identificación de las necesidades del negocio, así como la interacción con los usuarios funcionales. Por ese motivo, en este apartado se definen los actores del sistema, es decir, los usuarios que interactuarán con el sistema y qué usos le darán. A continuación, se muestra un diagrama de contexto para que se pueda visualizar de manera más sencilla (mediante un gráfico) la información que se ha descrito, y dar más información de que acciones llevará a cabo cada rol. Seguidamente, se muestra el diagrama de las historias de usuario, y para finalizar, se detallaran esas historias de usuario con su especificación textual. En la especificación textual podremos ver la descripción de las actividades entre el sistema y los actores para poder alcanzar los objetivos de éstos.

Para la especificación se ha decidido utilizar las historias de usuario. Hemos optado por este método en lugar de los casos de uso, ya que el sistema es autónomo y la mayoría funcionalidades son propias del sistema sin interactuar directamente con el usuario.

10.1 Actores

Cliente

Usuario que podrá ver los datos finales de los sensores para poder ver el estado de todos los módulos que están instalados en casa del paciente y así pueda consultar siempre que quiera el estado de la vivienda. Normalmente, éste usuario será un familiar directo del paciente. A estos usuarios son a los que les llegará la notificación en caso de evidenciar algún peligro en la casa del paciente.

Paciente

El paciente es el usuario en casa del cual se instalará el sistema de detección y monitorización. Éste usuario es importante, ya que en caso de detectar algún peligro, éste será el primer usuario al que se avisará, y si puede poner fin al peligro sin correr riesgos, se le dará la opción de apagar la alarma y ponerle solución a ese peligro. Ésta opción solo estará disponible si el peligro es pequeño, como una pequeña fuga de gas o una pequeña fuga de agua.

Servicios de emergencia

Usuario indirecto que no interactuará directamente con el sistema, sino que el sistema podrá ponerse en contacto con los servicios de emergencia en caso de detectar un peligro grave, como un incendio o una fuga grande de gas, y así poder actuar antes y con más efectividad y seguridad.

10.2 Diagrama de contexto

A continuación se mostrará el diagrama de contexto del sistema a desarrollar. En este diagrama veremos de manera gráfica como interactuarán los actores descritos anteriormente con nuestro sistema, y veremos que necesidades tiene cada rol.

Como se puede ver en el diagrama, el sistema es muy autónomo para facilitar su uso, y la mayor parte de funcionalidades son propias del sistema y las lleva a cabo autónomamente.

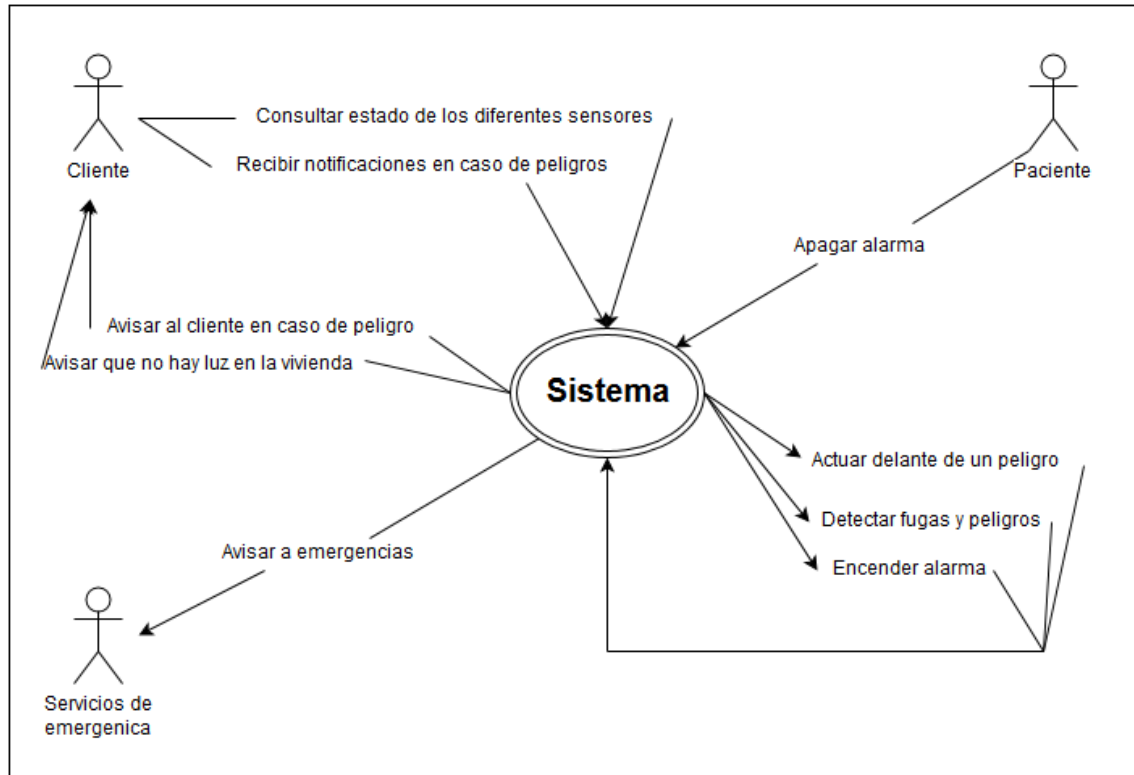


Fig. 4 Diagrama de contexto

10.3 Diagramas de historias de usuario

Primero de todo, vamos a definir las historias de usuario.

Las historias de usuario son cortas, simples descripciones de una funcionalidad descrita desde la perspectiva de la persona que quiere esa funcionalidad ^[18].

Normalmente siguen el siguiente esquema:

Como <tipo de usuario>, quiero <objetivo deseado> para que <alguna razón>.

El criterio más importante para escribir buenas historias de usuario es el INVEST ^[19]:

- **Independent (Independiente):** se pueden hacer en cualquier orden (no dependen unas de otras)
- **Negotiable (Negociable):** Se pueden reemplazar por otras de diferente prioridad.
- **Valuable (Valor):** Que sean necesarias y de valor para el proyecto.
- **Estimable:** Que se pueda estimar de manera segura y tranquila.
- **Small (Pequeñas):** que no sean grandes, funcionalidades pequeñas.
- **Testable (Verificable):** que se le puedan realizar pruebas.

A continuación os vamos a mostrar los diagramas de las historias de usuario. Para facilitar la visualización de estos diagramas, vamos a dividirlos en 3: un diagrama para el sistema, otro para la aplicación móvil y otro para la aplicación web.

Diagrama del sistema

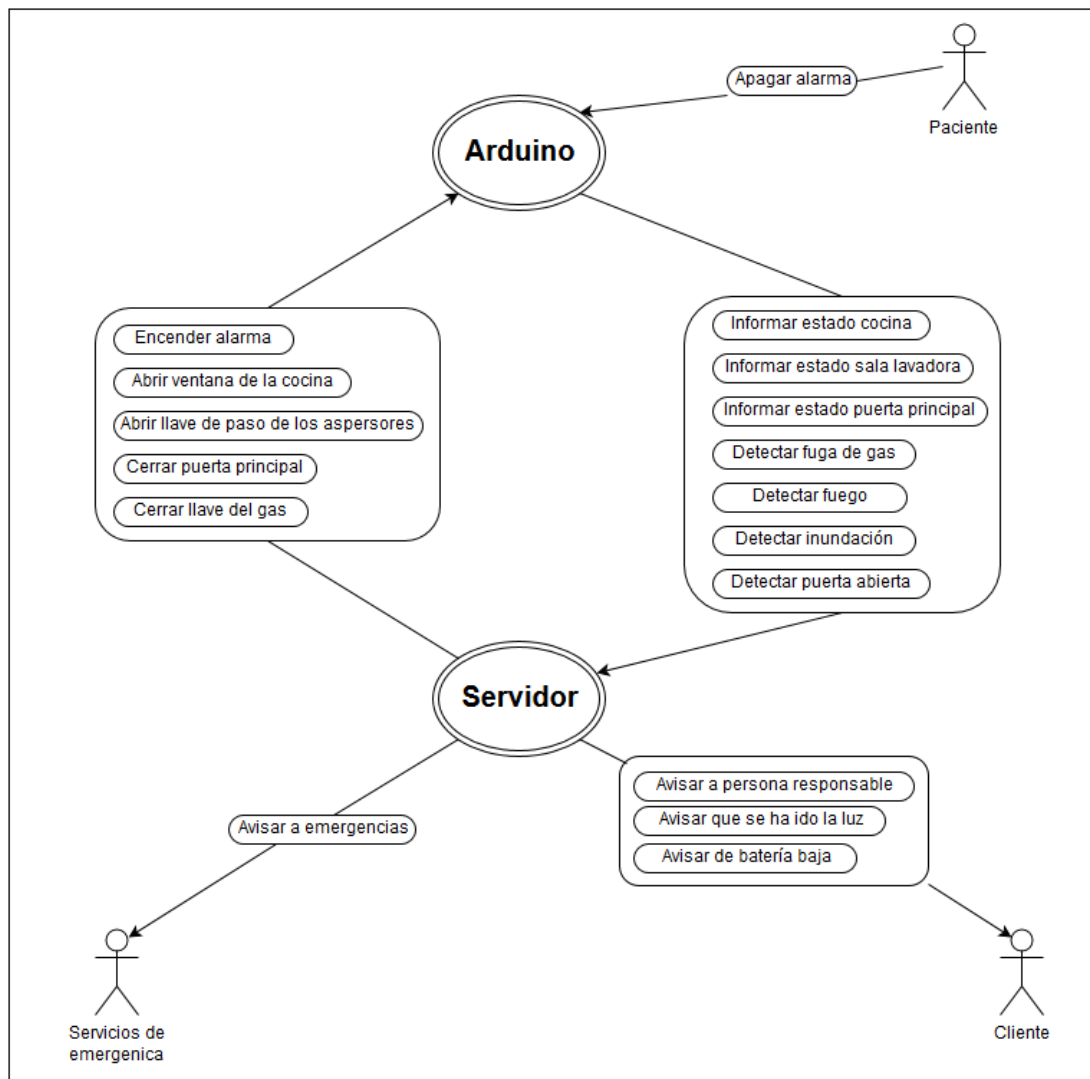


Fig. 5 Diagrama del sistema

Diagrama de la aplicación móvil

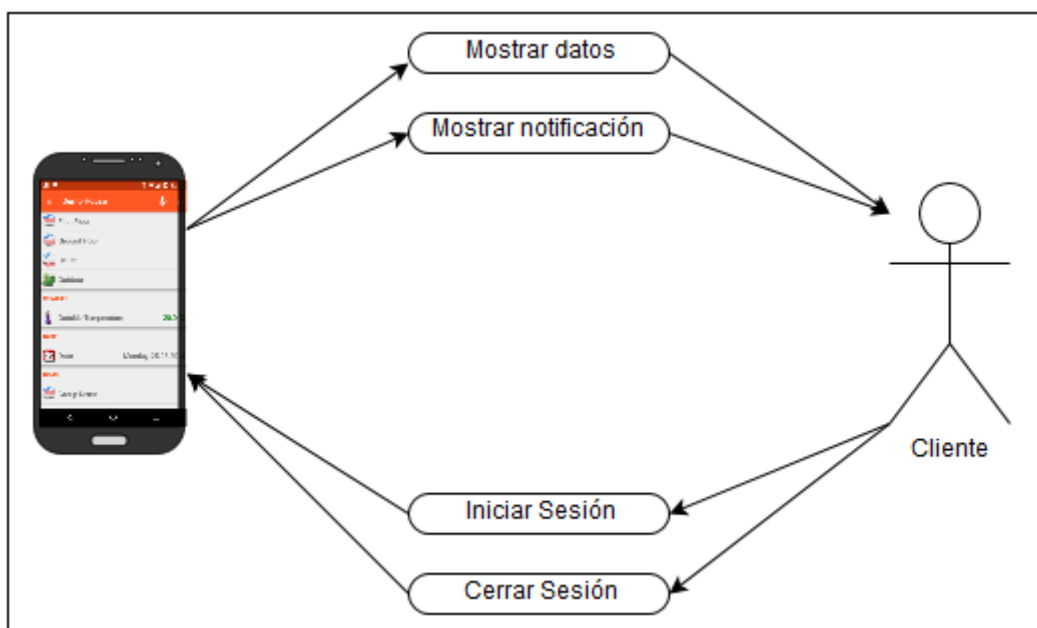


Fig. 6 Diagrama de la aplicación móvil

Diagrama de la aplicación web

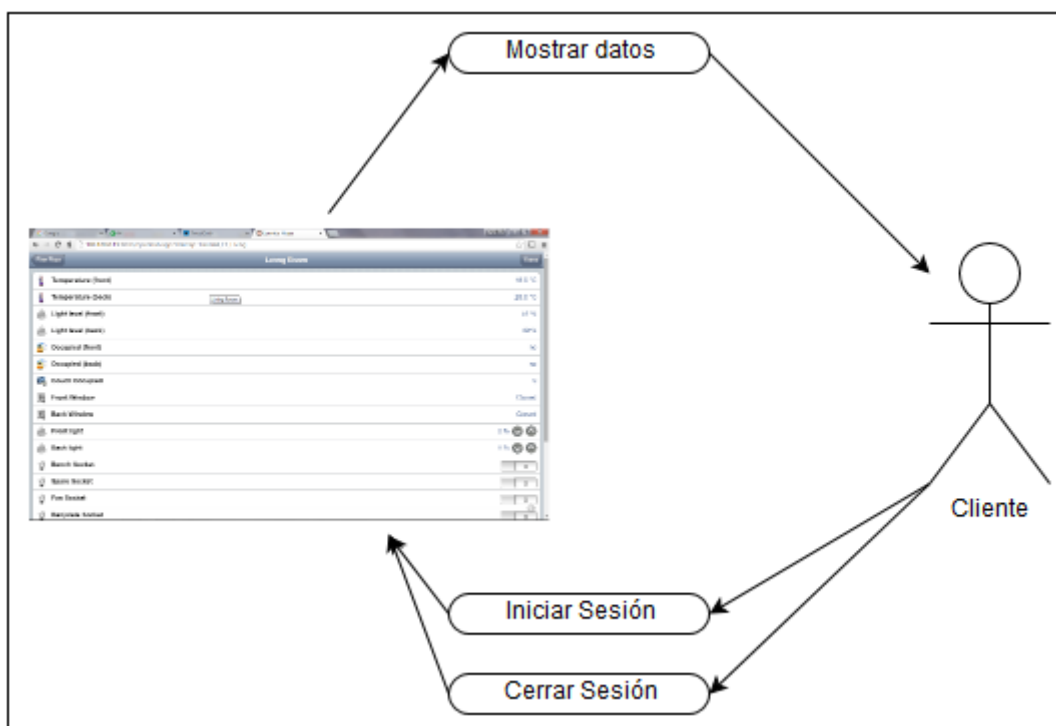


Fig. 7 Diagrama de la aplicación web

10.4 Especificación textual de las historias de usuario

A continuación definiremos las historias de usuario. Seguiremos una fácil estructura: primero definiremos la historia de usuario siguiendo su típica estructura:

Como *<tipo de usuario>*, quiero *<objetivo deseado>* para que *<alguna razón>*

Seguiremos la estructura para dejar bien claro qué tipo de usuario quiere esa funcionalidad, el objetivo de esa funcionalidad y por qué el usuario quiere la misma.

A continuación, escribiremos los criterios de aceptación de esa funcionalidad.

Historias de usuario del sistema

HU: 1 - Informar estado cocina

Como cliente

Quiero que el sistema informe del estado de la cocina

Para que pueda saber el estado de los sensores.

Criterios de aceptación

Criterio de aceptación 1: Informar cuando no hay peligro

- Cuando no haya ni fuego ni gas
- El sistema informará que no hay ni fuego ni gas

Criterio de aceptación 2: Informar cuando hay peligro de gas

- Cuando hay peligro de gas
- El sistema informará que hay peligro de gas
- El sistema informará si hay poco gas o mucho gas

Criterio de aceptación 3: Informar cuando hay peligro de fuego

- Cuando hay peligro de fuego
- El sistema informará que hay peligro de fuego
- El sistema informará si hay fuego cerca o fuego lejos

HU: 2 - Informar estado sala lavadora

Como cliente

Quiero que el sistema informe del estado de la sala de la lavadora

Para que pueda saber el estado de los sensores

Criterios de aceptación

Criterio de aceptación 1: Informar cuando no hay peligro

- Cuando no hay inundación
- El sistema informará que no hay peligro

Criterio de aceptación 2: Informar cuando hay peligro inundación

- Cuando hay peligro de inundación
- El sistema informará que hay peligro de inundación
- El sistema informará si hay poca agua o mucha agua.

HU: 3 - Informar estado puerta principal

Como cliente

Quiero que el sistema informe del estado de la puerta principal

Para que pueda saber el estado de los sensores

Criterios de aceptación

Criterio de aceptación 1: Informar cuando no hay peligro

- Cuando no está la puerta abierta
- El sistema informará que no hay peligro

Criterio de aceptación 2: Informar cuando hay peligro inundación

- Cuando esté la puerta abierta por unos segundos determinados
- El sistema informará que la puerta lleva abierta mucho tiempo
- El sistema informará el tiempo que lleva la puerta abierta.

HU: 4 - Detectar fuga de gas

Como cliente

Quiero que el sistema detecte si hay una fuga de gas

Para que pueda saber que hay una fuga de gas en la cocina.

Criterios de aceptación

Criterio de aceptación 1: Detectar fuga de gas

- Cuando haya una fuga de gas en la cocina
- El sistema detectará que hay una fuga de gas
- El sistema detectará si hay poco gas o mucho gas

HU: 5 - Detectar fuego

Como cliente

Quiero que el sistema detecte si hay fuego

Para que pueda saber que hay fuego en la cocina.

Criterios de aceptación

Criterio de aceptación 1: Detectar fuego

- Cuando haya fuego en la cocina
- El sistema detectará que hay fuego
- El sistema detectará si hay fuego cerca o fuego lejos

HU: 6 - Detectar inundación

Como cliente

Quiero que el sistema detecte si hay una inundación

Para que pueda saber que hay una inundación en la sala de la lavadora.

Criterios de aceptación

Criterio de aceptación 1: Detectar inundación

- Cuando haya agua en la sala de la lavadora
- El sistema detectará que hay una inundación
- El sistema detectará si hay mucha agua o poca agua.

HU: 7 - Detectar puerta abierta

Como cliente

Quiero que el sistema detecte si está la puerta principal abierta

Para que pueda saber si la puerta principal se ha quedado abierta demasiado tiempo.

Criterios de aceptación

Criterio de aceptación 1: Detectar puerta abierta

- Cuando la puerta principal esté abierta demasiado tiempo
- El sistema detectará que se ha dejado la puerta abierta
- El sistema detectará cuánto tiempo lleva la puerta abierta.

HU: 8 - Encender alarma

Como cliente

Quiero que el sistema encienda una alarma si hay algún peligro

Para que el paciente esté informado del peligro y pueda solucionar el problema rápidamente si se trata de un problema leve

Criterios de aceptación

Criterio de aceptación 1: Encender alarma

- Cuando haya un peligro
- El sistema encenderá la alarma

HU: 9 - Apagar alarma

Como paciente

Quiero que el sistema apague la alarma cuando apreté el botón de apagar la alarma

Para que sepa que estoy solucionando el problema y deje de hacer ruido.

Criterios de aceptación

Criterio de aceptación 1: Apagar alarma

- Cuando esté la alarma encendida
- Cuando el paciente apriete el botón de apagar la alarma
- Cuando el peligro es pequeño
- El sistema apagará la alarma

Criterio de aceptación 2: No apagar alarma

- Cuando esté la alarma encendida
- Cuando el paciente apriete el botón de apagar la alarma
- Cuando el peligro sea grande
- El sistema no apagará la alarma

HU: 10 - Avisar a persona responsable

Como cliente

Quiero que el sistema me avise si hay algún peligro en la casa del paciente

Para que pueda saber del peligro y actuar en consecuencia.

Criterios de aceptación

Criterio de aceptación 1: Avisar a persona responsable

- Cuando haya un peligro en la casa
- Avisar a la persona responsable del paciente

HU: 11 - Avisar a emergencias

Como cliente

Quiero que el sistema avise a emergencias en caso de detectar un peligro grave

Para que se pueda actuar rápido y evitar daños del paciente o daños materiales en la vivienda.

Criterios de aceptación

Criterio de aceptación 1: Avisar a emergencias con peligro grande

- Cuando haya un peligro grande en la casa
- Avisar a emergencias

Criterio de aceptación 2: Avisar a emergencias con peligro pequeño

- Cuando haya un peligro pequeño en la casa
- Cuando la alarma no haya sido apagada en 1 minuto
- Avisar a emergencias

Criterio de aceptación 3: no avisar a emergencias con peligro pequeño y alarma apagada

- Cuando haya un peligro pequeño y en la casa
- Cuando la alarma haya sido apagada en menos de 1 minuto
- No avisar a emergencias

HU: 12 - Avisar que se ha ido la luz

Como cliente

Quiero que se me avise si se va la luz

Para que el sistema no esté desactivado y no pueda detectar peligros

Criterios de aceptación

Criterio de aceptación 1: Avisar que se ha ido la luz

- Cuando se vaya la luz
- Avisar que se ha ido la luz

Criterio de aceptación 2: Avisar que no hay luz cada 30 minutos

- Cuando no haya luz
- Cuando haya pasado 30 minutos desde el último aviso
- Avisar el tiempo que el sistema lleva sin luz

HU: 13 - Avisar de batería baja

Como cliente

Quiero que se me avise cuando un módulo tenga poca batería

Para que pueda reemplazarla y no se agote la batería.

Criterios de aceptación

Criterio de aceptación 1: Avisar de batería baja

- Cuando la batería de un módulo esté al 25%
- Avisar a la persona responsable del paciente de batería baja en ese módulo.

Criterio de aceptación 2: Avisar de batería muy baja

- Cuando la batería de un módulo esté al 15%
- Avisar a la persona responsable del paciente de batería muy baja en ese módulo.

Criterio de aceptación 3: Avisar que el módulo está a punto de apagarse

- Cuando la batería de un módulo esté al 5%
- Avisar a la persona responsable del paciente de que ese módulo está a punto de apagarse

HU: 14 - Abrir ventana de la cocina

Como cliente

Quiero que se abra la ventana de la cocina si se detecta fuga de gas

Para que se escape el gas por la ventana y se desvanezca el peligro

Criterios de aceptación

Criterio de aceptación 1: Abrir ventana de la cocina

- Cuando se detecte una fuga de gas en la cocina
- Abrir la ventana de la cocina.

HU: 15 - Abrir llave de paso de los aspersores

Como cliente

Quiero que se abra la llave de paso de los aspersores si se detecta fuego en la cocina

Para que se pueda extinguir el fuego de la cocina.

Criterios de aceptación

Criterio de aceptación 1: Abrir llave de paso de los aspersores

- Cuando se detecte fuego en la cocina
- Abrir llave de paso de los aspersores

HU: 16 - Cerrar llave del gas

Como cliente

Quiero que se cierre la llave de paso del gas cuando haya fuego o gas en la cocina

Para que si el paciente se ha dejado el gas abierto o el fuego encendido, se pueda solucionar el peligro.

Criterios de aceptación

Criterio de aceptación 1: Cerrar llave de paso del gas cuando haya gas

- Cuando se detecte una fuga de gas en la cocina
- Cerrar la llave de paso del gas

Criterio de aceptación 2: Cerrar llave del gas cuando haya fuego

- Cuando se detecte fuego en la cocina
- Cerrar la llave de paso del gas

HU: 17 - Cerrar puerta principal

Como cliente

Quiero que se cierre la puerta principal de la casa del paciente si lleva demasiado tiempo abierta

Para que no se pueda quedar abierta si al paciente se le olvida cerrarla o no la cierra bien.

Criterios de aceptación

Criterio de aceptación 1: Cerrar puerta principal

- Cuando la puerta principal lleve 1 minuto abierta
- Cuando la alarma no haya sido parada
- Cerrar la puerta principal

Criterio de aceptación 2: No cerrar puerta principal

- Cuando la puerta principal lleve 1 minuto abierta
- Cuando la alarma haya sido parada
- No Cerrar la puerta principal

Historias de usuario de la aplicación móvil

HU: 18 - Iniciar Sesión

Como cliente

Quiero iniciar sesión en la aplicación móvil

Para que pueda ver el estado de los sensores de la casa de mi paciente y pueda recibir notificaciones y avisos de los posibles peligros de la casa desde la aplicación móvil.

Criterios de aceptación

Criterio de aceptación 1: Iniciar sesión exitoso

- Cuando el ingreso del usuario y contraseña son correctos
- El sistema permitirá el ingreso a la aplicación móvil

Criterio de aceptación 2: Iniciar sesión fallido

- Cuando el ingreso del usuario y contraseña son incorrectos
- El sistema no permitirá el ingreso a la aplicación móvil
- El sistema presentará una alerta con un mensaje para informar que el usuario y/o la contraseña no son válidos.

HU: 19 - Cerrar Sesión

Como cliente

Quiero cerrar sesión en la aplicación móvil

Para que pueda cerrar mi sesión en la aplicación móvil

Criterios de aceptación

Criterio de aceptación 1: Cerrar sesión

- Cuando le indique al sistema que quiero cerrar la sesión
- El sistema cerrará la sesión del usuario en la aplicación móvil

HU: 20 - Mostrar datos

Como cliente

Quiero que la aplicación móvil me muestre el estado de los sensores de la casa de mi paciente

Para que pueda consultar el estado de los sensores desde el móvil

Criterios de aceptación

Criterio de aceptación 1: Mostrar datos

- Cuando se abra la aplicación móvil
- Mostrar los datos de los sensores y de las últimas alarmas

HU: 21 - Mostrar notificación

Como cliente

Quiero recibir una notificación en mi móvil cada vez que se detecte un peligro

Para que pueda saber que hay un peligro en la casa del paciente sin tener que estar mirando la aplicación móvil todo el día.

Criterios de aceptación

Criterio de aceptación 1: Mostrar notificación

- Cuando haya que avisar a la persona responsable de algún peligro o aviso
- Se mostrará una notificación en el móvil del cliente

Historias de usuario de la web

HU: 22 - Iniciar Sesión

Como cliente

Quiero iniciar sesión en la aplicación web

Para que pueda ver el estado de los sensores de la casa de mi paciente desde la aplicación web.

Criterios de aceptación

Criterio de aceptación 1: Iniciar sesión exitoso

- Cuando el ingreso del usuario y contraseña son correctos
- El sistema permitirá el ingreso a la aplicación web

Criterio de aceptación 2: Iniciar sesión fallido

- Cuando el ingreso del usuario y contraseña son incorrectos
- El sistema no permitirá el ingreso a la aplicación web
- El sistema presentará una alerta con un mensaje para informar que el usuario y/o la contraseña no son válidos.

HU: 23 - Cerrar Sesión

Como cliente

Quiero cerrar sesión en la aplicación web

Para que pueda cerrar mi sesión en la aplicación web

Criterios de aceptación

Criterio de aceptación 1: Cerrar sesión

- Cuando le indique al sistema que quiero cerrar la sesión
- El sistema cerrará la sesión del usuario en la aplicación web

HU: 24 - Mostrar datos

Como cliente

Quiero que la aplicación web me muestre el estado de los sensores de la casa de mi paciente

Para que pueda consultar el estado de los sensores desde una página web

Criterios de aceptación

Criterio de aceptación 1: Mostrar datos

- Cuando se abra la aplicación web
- Mostrar los datos de los sensores y de las últimas alarmas

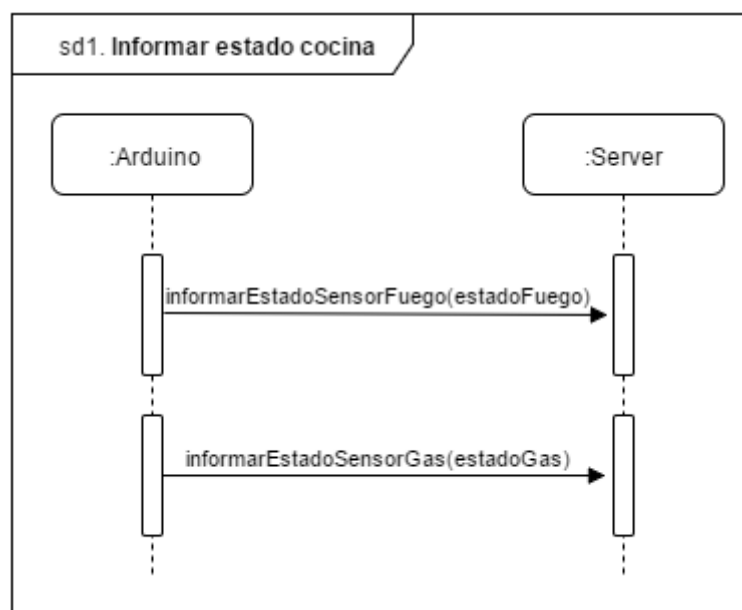
10.5 Modelo de comportamiento

El modelo de comportamiento tiene como objetivo elaborar los diagramas de secuencia entre el sistema y el actor, para determinar y especificar como es la interacción entre el actor y el sistema en cada una de las historias de usuario definidas anteriormente.

Seguiremos el modelo utilizado en los apartados anteriores y separaremos los diagramas de secuencia en tres apartaos: el primero para los diagramas de secuencia del sistema, el segundo para los diagramas de secuencia de la aplicación móvil, y el tercero y último para los diagramas de secuencia de la aplicación web.

Diagramas de secuencia del sistema

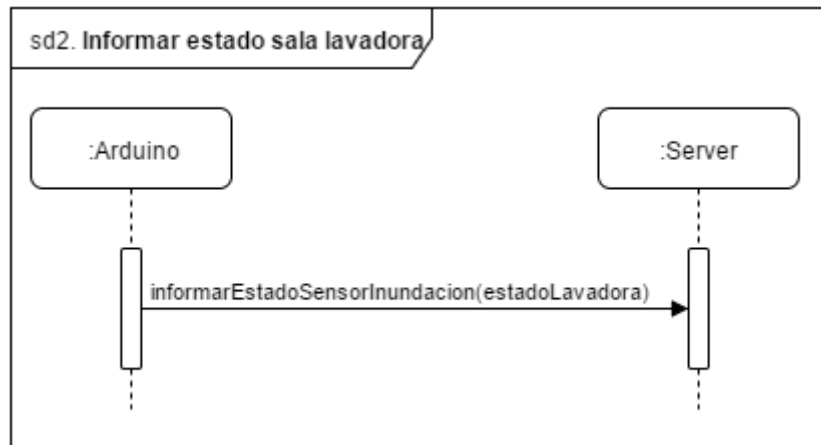
HU: 1 - Informar estado cocina



- **Contexto:** informarEstadoSensorFuego(estadoFuego:Stirng)
- **Pre:** *estadoFuego* no está vacío.
- **Post:** El servidor obtiene la información del estado del sensor de fuego

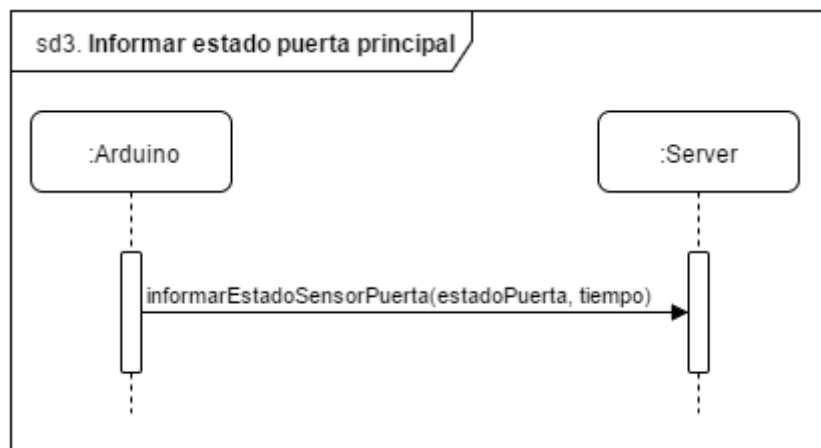
- **Contexto:** informarEstadoSensorGas(estadoGas:Stirng)
- **Pre:** *estadoGas* no está vacío.
- **Post:** El servidor obtiene la información del estado del sensor de gas

HU: 2 - Informar estado sala lavadora



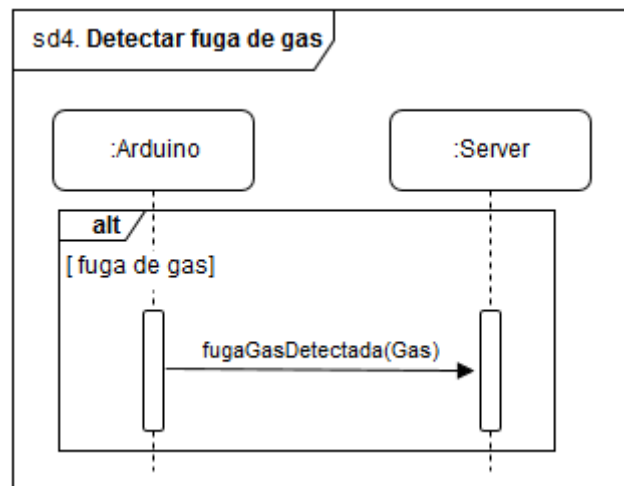
- **Contexto:** `informarEstadoSensorInundacion(estadoLavadora:Stirng)`
- **Pre:** `estadoLavadora` no está vacío.
- **Post:** El servidor obtiene la información del estado del sensor de inundación

HU: 3 - Informar estado puerta principal



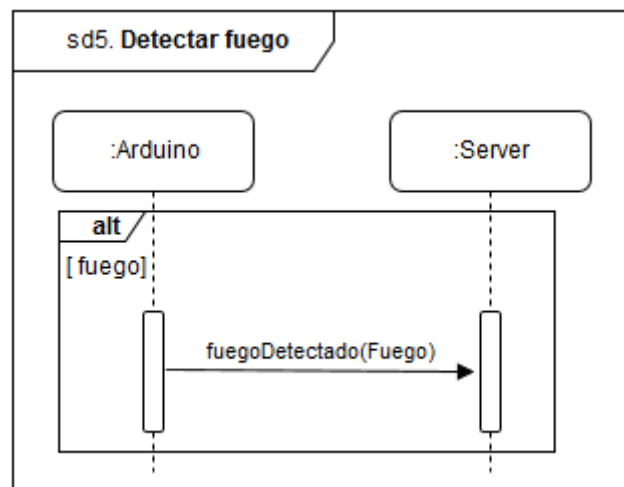
- **Contexto:** `informarEstadoSensorPuerta(estadoPuerta:Stirng, tiempo:Long)`
- **Pre:** `estadoPuerta` no está vacío.
`tiempo` no está vacío.
- **Post:** El servidor obtiene la información del estado del sensor de la puerta

HU: 4 - Detectar fuga de gas



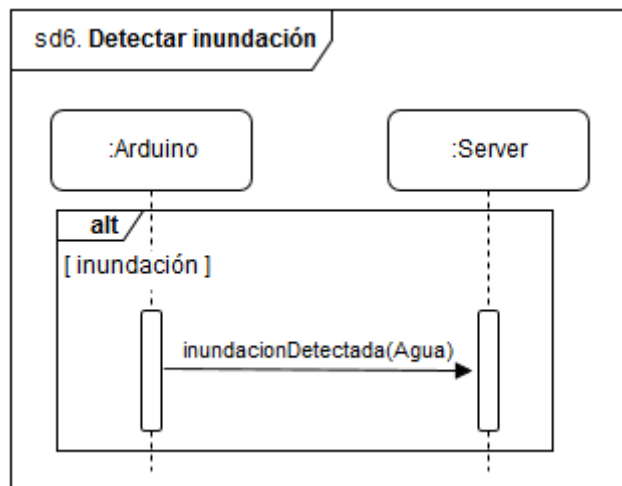
- **Contexto:** fugaGasDetectada(Gas:String)
- **Pre:** Gas no está vacío
Hay una fuga de gas
- **Post:** El servidor es informado que hay una fuga de gas, y de la cantidad de gas

HU: 5 - Detectar fuego



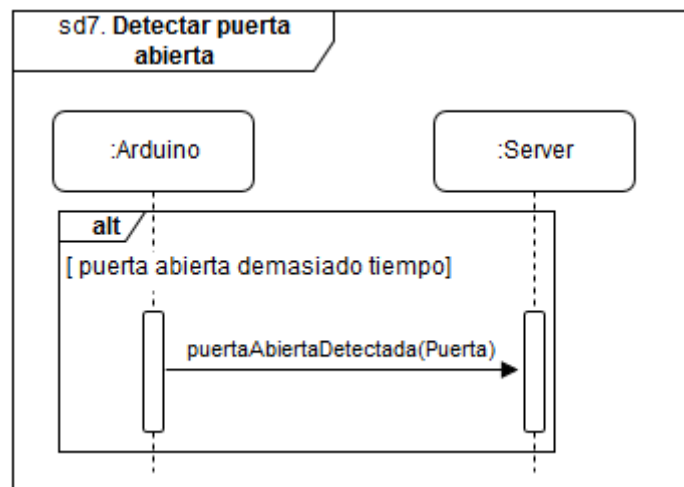
- **Contexto:** fuegoDetectado(Fuego:String)
- **Pre:** Fuego no está vacío
Hay fuego
- **Post:** El servidor es informado que hay fuego, y de la cantidad de fuego

HU: 6 - Detectar inundación



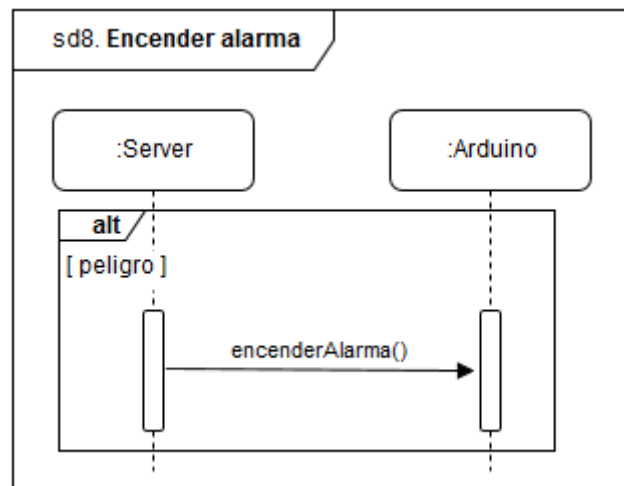
- **Contexto:** inundacionDetectada(Agua:String)
- **Pre:** *Agua* no está vacío
Hay una inundación
- **Post:** El servidor es informado que hay una inundación, y de la cantidad de agua

HU: 7 - Detectar puerta abierta



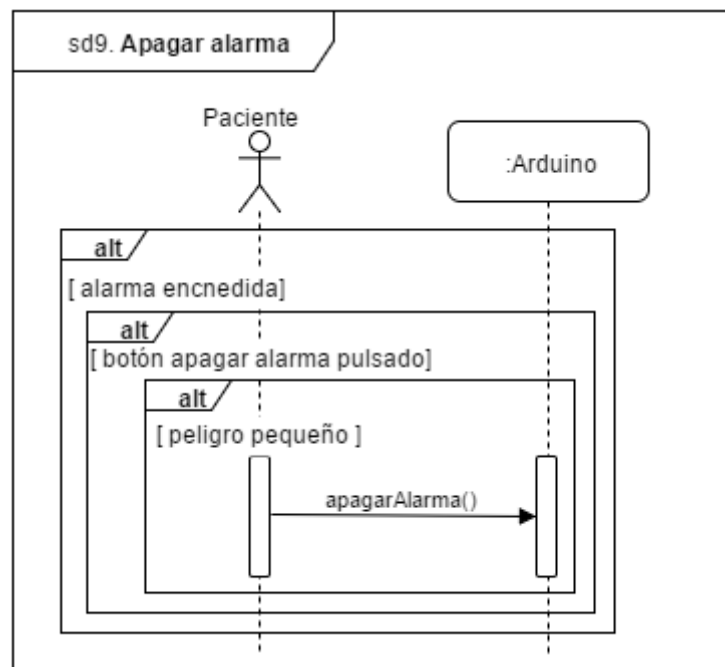
- **Contexto:** puertaAbiertaDetectada(Puerta:String)
- **Pre:** *Puerta* no está vacío
La puerta lleva demasiado tiempo abierta
- **Post:** El servidor es informado que la puerta lleva demasiado tiempo abierta, y el tiempo que lleva abierta.

HU: 8 - Encender alarma



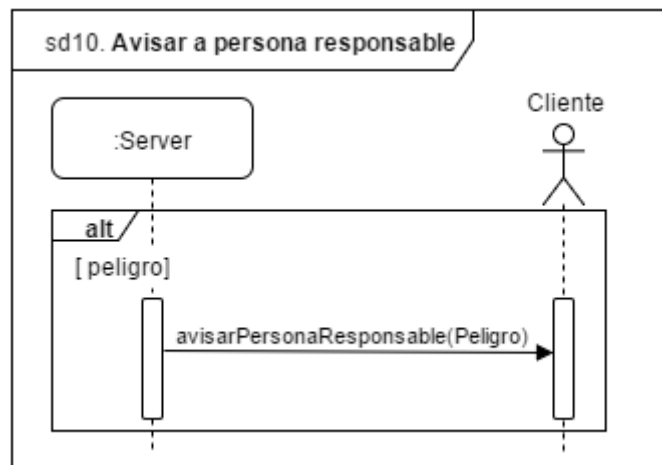
- **Contexto:** encenderAlarma()
- **Pre:** Se ha detectado un peligro
- **Post:** Se enciende la alarma de la casa

HU: 9 - Apagar alarma



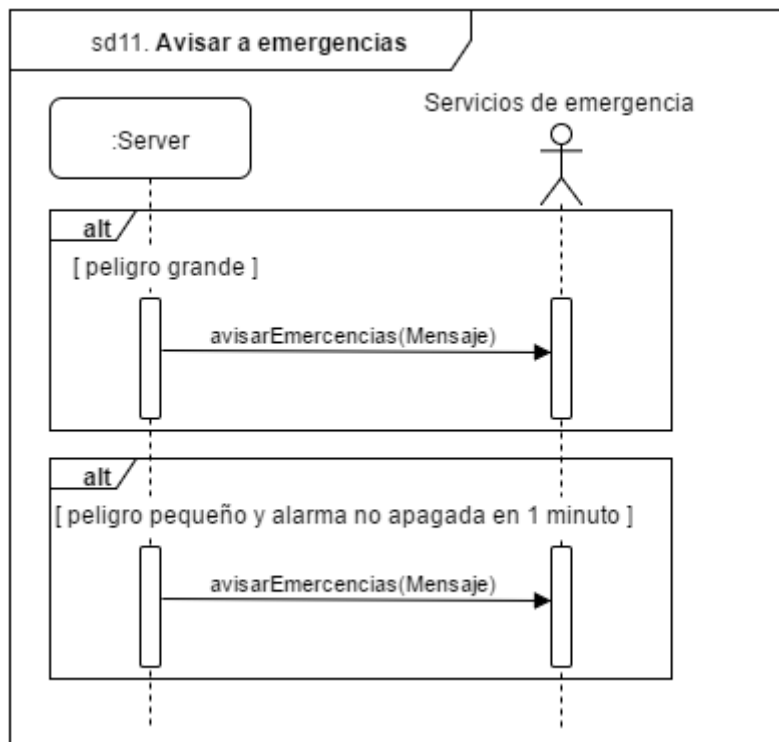
- **Contexto:** apagarAlarma()
- **Pre:** La alarma está encendida
El botón de apagar la alarma ha sido pulsado
El peligro es pequeño
- **Post:** Se apaga la alarma de la casa

HU: 10 - Avisar a persona responsable



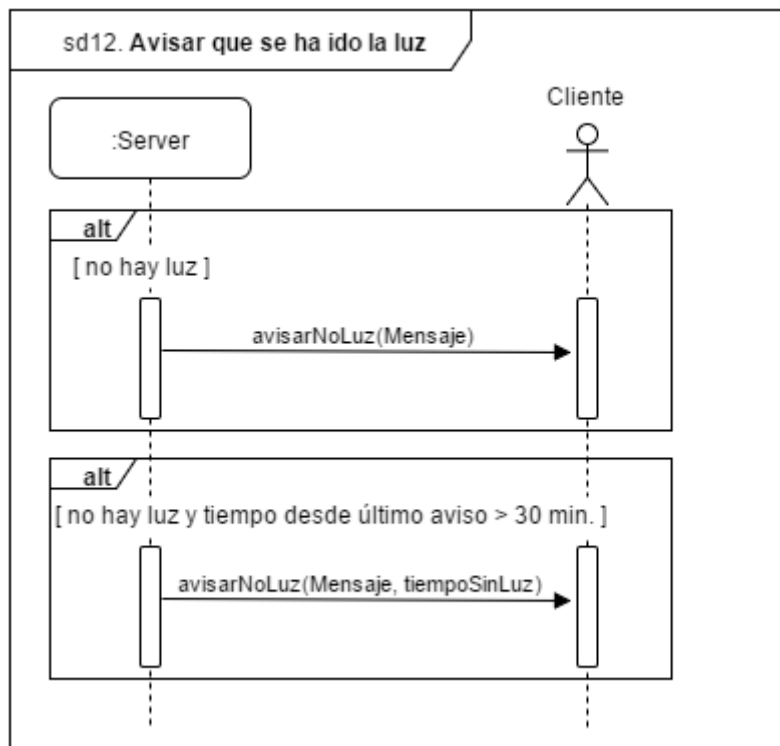
- **Contexto:** avisarPersonaResponsable(Peligro:String)
- **Pre:** *Peligro* no está vacío
Existe un peligro en la casa
- **Post:** Se avisará a la persona responsable que hay un peligro en la casa mediante un SMS o una notificación en su teléfono móvil

HU: 11 - Avisar a emergencias



- **Contexto:** avisarEmergencias(Mensaje)
- **Pre:** *Mensaje* no está vacío
Cuando se cumple una de las siguientes condiciones:
 - Cuando hay un peligro grande
 - Cuando hay un peligro pequeño y la alarma no se ha apagado en 1 minuto o menos
- **Post:** Se avisa a emergencias llamando al 112 transmitiendo el mensaje *Mensaje*

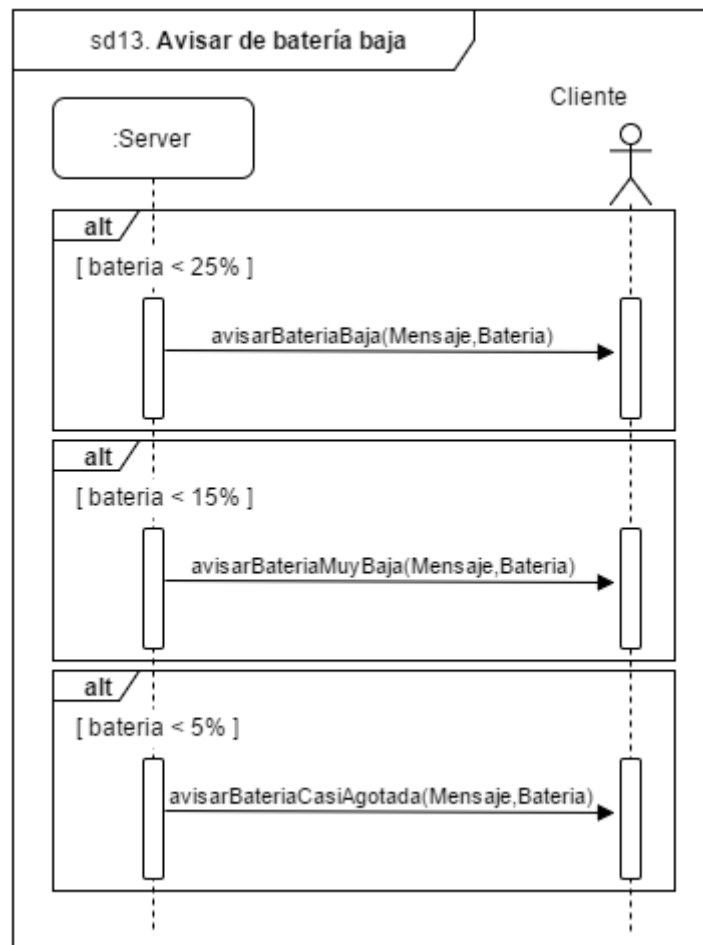
HU: 12 - Avisar que se ha ido la luz



- **Contexto:** avisarNoLuz(Mensaje:String)
- **Pre:** *Mensaje* no está vacío
No hay luz en la casa
- **Post:** Se avisará a la persona responsable no hay luz en la casa mediante un SMS o una notificación en su teléfono móvil

- **Contexto:** avisarNoLuz(Mensaje:String, tiempoSinLuz:Long)
- **Pre:** *Mensaje* no está vacío
No hay luz en la casa
Han pasado más de 30 minutos desde el último aviso
- **Post:** Se avisará a la persona responsable no hay luz en la casa y el tiempo que lleva sin luz la casa mediante un SMS o una notificación en su teléfono móvil

HU: 13 - Avisar de batería baja

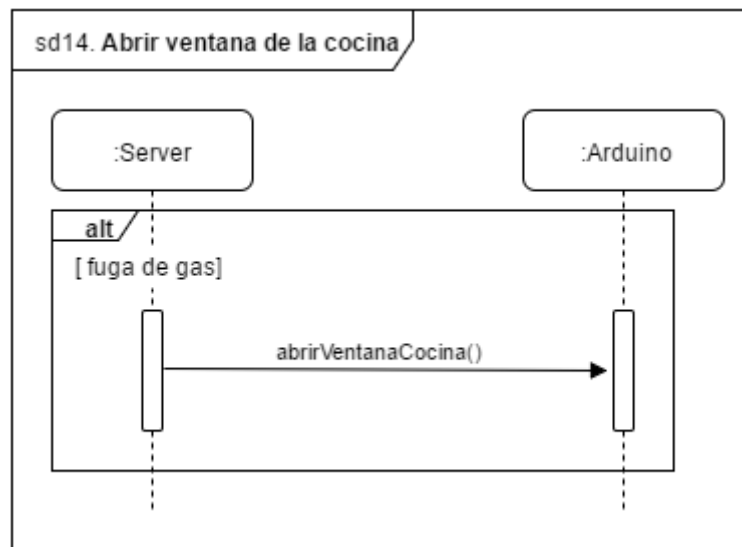


- **Contexto:** avisarBateriaBaja(Mensaje:String, Bateria:Integer)
- **Pre:**
 - Mensaje* no está vacío
 - Bateria* no está vacío
 - Batería del módulo por debajo del 25%
- **Post:** Se avisará a la persona responsable que el módulo tiene poca batería y la batería que queda mediante un SMS o una notificación en su teléfono móvil

- **Contexto:** avisarBateriaMuyBaja(Mensaje:String, Bateria:Integer)
- **Pre:**
 - Mensaje* no está vacío
 - Bateria* no está vacío
 - Batería del módulo por debajo del 15%
- **Post:** Se avisará a la persona responsable que el módulo tiene poca batería y la batería que queda mediante un SMS o una notificación en su teléfono móvil

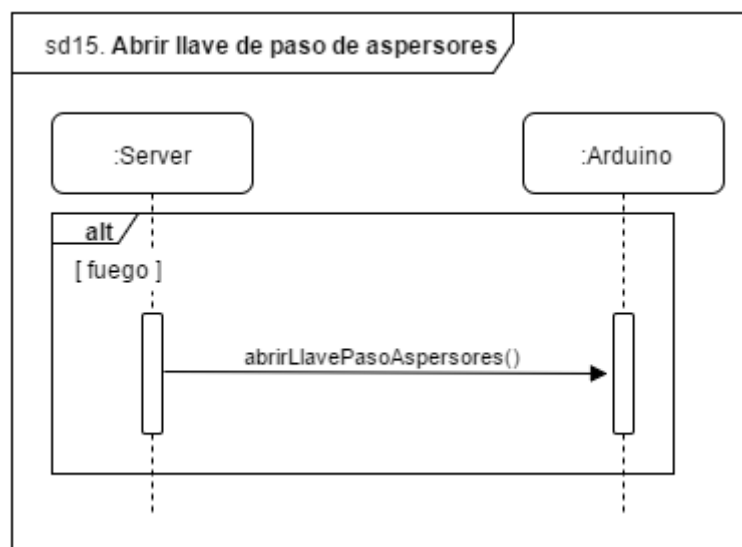
- **Contexto:** avisarBateriaCasiAgotada (Mensaje:String, Bateria:Integer)
- **Pre:**
 - Mensaje* no está vacío
 - Bateria* no está vacío
 - Batería del módulo por debajo del 5%
- **Post:** Se avisará a la persona responsable que el módulo tiene poca batería y la batería que queda mediante un SMS o una notificación en su teléfono móvil

HU: 14 - Abrir ventana de la cocina



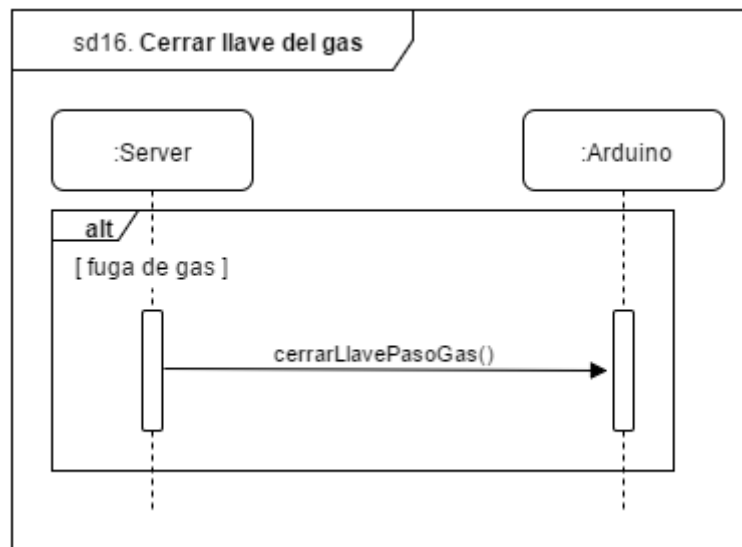
- **Contexto:** abrirVentanaCocina()
- **Pre:** Hay una fuga de gas
- **Post:** Se abrirá la ventana de la cocina

HU: 15 - Abrir llave de paso de los aspersores



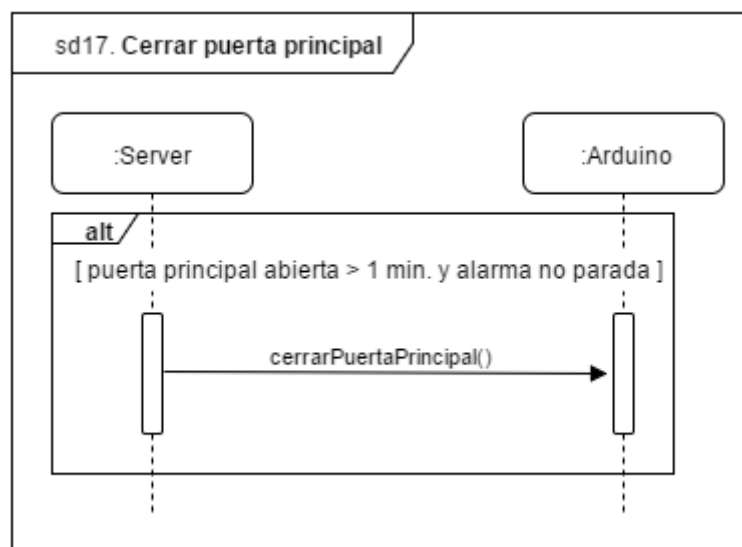
- **Contexto:** abrirLlavePasoAspersores()
- **Pre:** Hay fuego
- **Post:** Se abrirá la llave de paso de los aspersores de la cocina

HU: 16 - Cerrar llave del gas



- **Contexto:** cerrarLlavePasoGas()
- **Pre:** Hay una fuga de gas
- **Post:** Se cerrará la llave de paso del gas

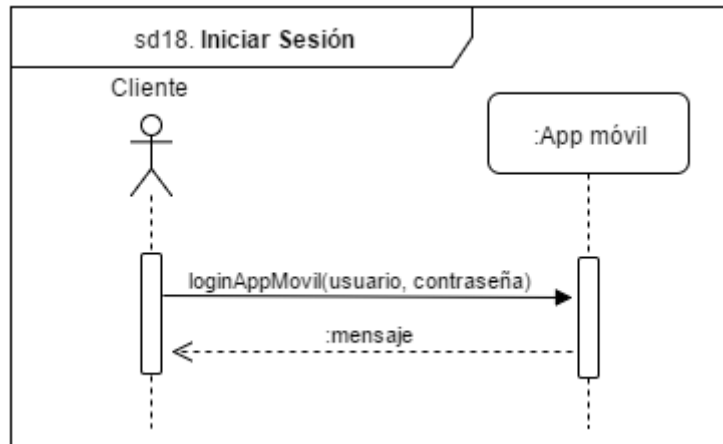
HU: 17 - Cerrar puerta principal



- **Contexto:** cerrarPuertaPrincipal()
- **Pre:** Puerta principal abierta más de 1 minuto
La alarma no ha sido parada
- **Post:** Se cerrará la puerta principal

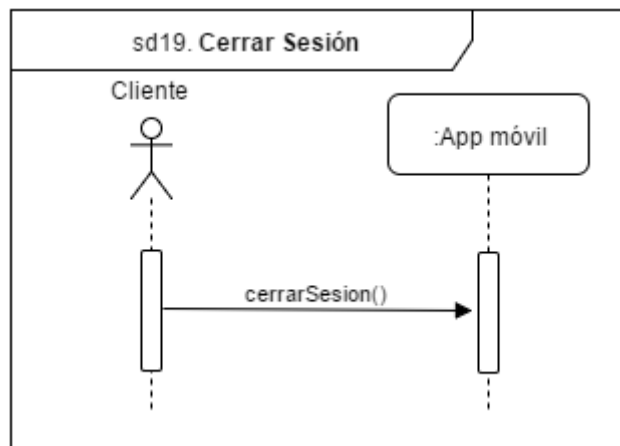
Diagramas de secuencia de la aplicación móvil

HU: 18 - Iniciar Sesión



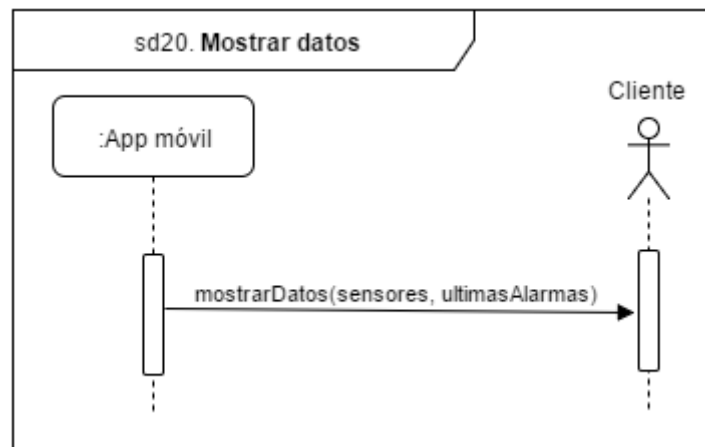
- **Contexto:** loginAppMovil(usuario:String, contraseña:String)
- **Pre:** *usuario* no está vacío
contraseña no está vacío
- **Post:** Si existe una cuenta de usuario con usuario *usuario* y contraseña *contraseña*, el usuario habrá iniciado la sesión. Si los datos no son correctos, el sistema presentará una alerta con el mensaje *mensaje* para informar que el usuario y/o la contraseña no son válidos.

HU: 19 - Cerrar Sesión



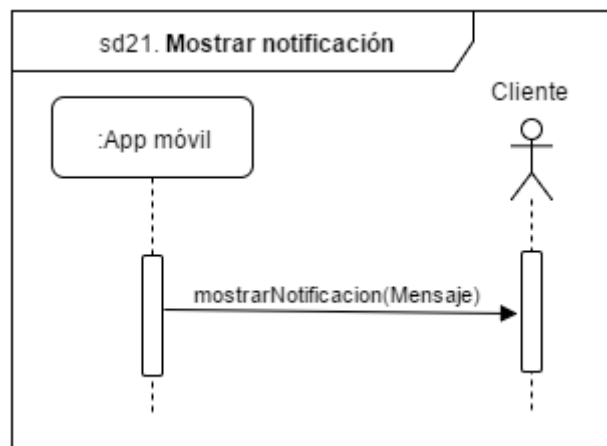
- **Contexto:** cerrarSesion()
- **Pre:** El usuario estaba logado en la aplicación móvil
- **Post:** Se cierra la sesión del usuario en la aplicación móvil

HU: 20 - Mostrar datos



- **Contexto:** mostrarDatos(sensores:Set(sensor:String),ultimasAlarmas:Set(alarma:String))
- **Pre:** *sensores* no está vacío
- **Post:** Se mostrarán en la aplicación móvil el estado de los sensores y las últimas alarmas detectadas

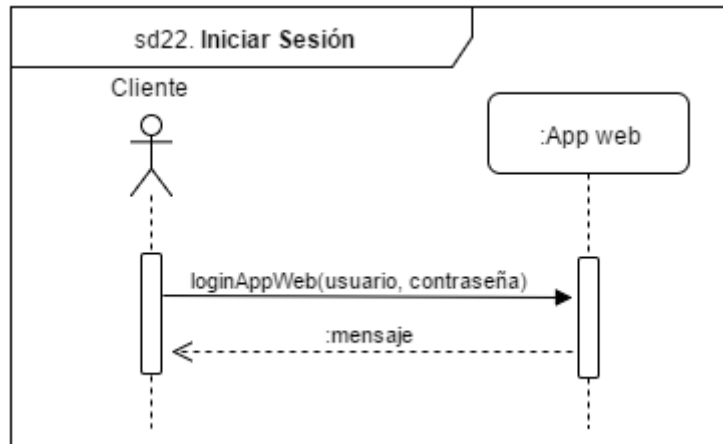
HU: 21 - Mostrar notificación



- **Contexto:** mostrarNotificacion(Mensaje:String)
- **Pre:** *Mensaje* no está vacío
- **Post:** Se mostrará una notificación al usuario con el mensaje *Mensaje*

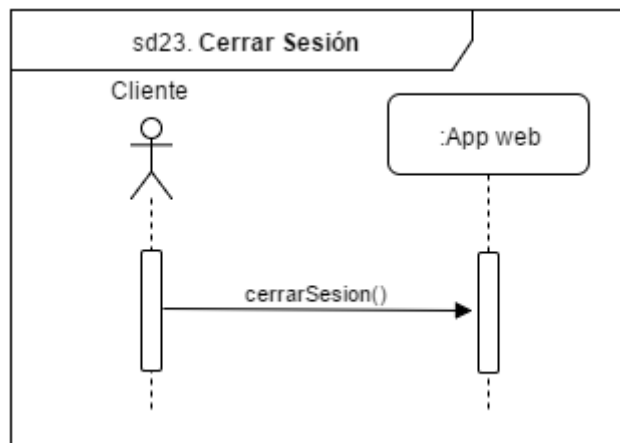
Diagramas de secuencia de la aplicación web

HU: 22 - Iniciar Sesión



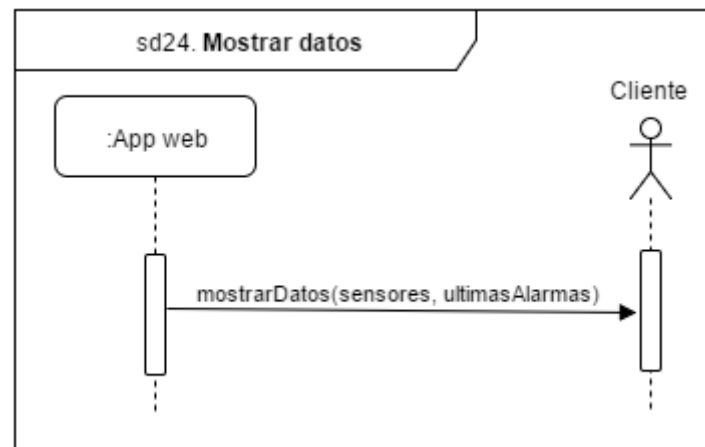
- **Contexto:** loginAppWeb(usuario:Stirng,contraseña:String)
- **Pre:** *usuario* no está vacío
contraseña no está vacío
- **Post:** Si existe una cuenta de usuario con usuario *usuario* y contraseña *contraseña*, el usuario habrá iniciado la sesión. Si los datos no son correctos, el sistema presentará una alerta con el mensaje *mensaje* para informar que el usuario y/o la contraseña no son válidos.

HU: 23 - Cerrar Sesión



- **Contexto:** cerrarSesion()
- **Pre:** El usuario estaba logado en la aplicación web
- **Post:** Se cierra la sesión del usuario en la aplicación web

HU: 24 - Mostrar datos



- **Contexto:** `mostrarDatos(sensores:Set(sensor:String),ultimasAlarmas:Set(alarma:String))`
- **Pre:** *sensores* no está vacío
- **Post:** Se mostrarán en la aplicación web el estado de los sensores y las últimas alarmas detectadas

11. DISEÑO DEL SERVIDOR

El servidor de nuestro sistema ha sido implementado en una Raspberry Pi 3 Modelo B. Primero de todo, vamos a hablar de ésta.

Que es Raspberry Pi?

Raspberry Pi ^[20] es una pequeña placa computadora de bajo coste desarrollada en Reino Unido por la Fundación Raspberry Pi con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas, es tan pequeño como una tarjeta de crédito, y funciona con un mínimo consumo eléctrico.

Éste pequeño ordenador es excelente para desarrollar software, es una herramienta ideal para aprender programación, y administración de sistemas informáticos y redes.

Raspberry Pi 3 Modelo B

La *Raspberry Pi 3 modelo B* es la última generación de *Raspberry Pi*. Es 10 veces más potente que la *Raspberry Pi* original y tiene una conectividad integrada de 802.11 b/g/n LAN y Bluetooth 4.1 (clásico y LE).

Además, su procesador Cortex-A7 de cuatro núcleos mejora pasando de 900 MHz a 1.2GHz. Como los demás modelos, funciona con la última versión de NOOBS ^[21] instalado en una tarjeta micro SD.

Está es perfectamente compatible con las versiones 1 y 2.



Fig. 8 Raspberry Pi

Características técnicas

- Dimensiones: 86 x 56 x 17 mm
- Procesador Broadcom BCM2837 de 1.2GHz ARM Cortex-A7 de cuatro núcleos con GPU VideoCore IV de doble núcleo
- GPU con tecnología Open GL ES 2.0, hardware acelerado OpenVG y admite imágenes de alta resolución 1080p30 H.264
- GPU con capacidad de 1 Gpixel/s, 1,5 Gtexel/s o 24 GFLOPs con filtrado e infraestructura DMA
- SDRAM LPDDR2 de 1 GB
- Salida de vídeo HD 1080p
- Salida de vídeo compuesto (PAL/NTSC)
- Salida de audio estéreo
- Conector hembra Ethernet RJ45 10/100 BaseT
- Conector hembra de vídeo/audio HDMI 1.3 y 1.4
- Conector hembra de salida de vídeo compuesto/audio de 3,5 mm 4 polos
- 4 conectores hembra USB 2.0
- Conector MPI CSI-2 de 15 vías para cámara de vídeo HD Raspberry Pi (775-7731)
- Conector de interfaz serie de display de 15 vías
- Conector para tarjeta MicroSD
- Arranca desde tarjeta MicroSD, ejecutando una nueva versión del sistema operativo Linux
- Conector macho de 40 pines para buses serie y GPIO (compatible con el conector macho de 26 pines Raspberry Pi 1)
- Fuente de alimentación: +5 V a 2 A a través de conector hembra microUSB

Conexiones

El dispositivo incluye conectores de expansión GPIO que pueden ser utilizados para la comunicación con otros dispositivos como módulos específicos de expansión, e incluso es compatible con Arduino, con lo que permite implementar un gran abanico de proyectos.

Además con sus pines GND y +5V se pueden alimentar dispositivos como por ejemplo ventiladores, y posee dos puertos de expansión específicos donde podemos conectar periféricos para la *Raspberry Pi* como su cámara HD.

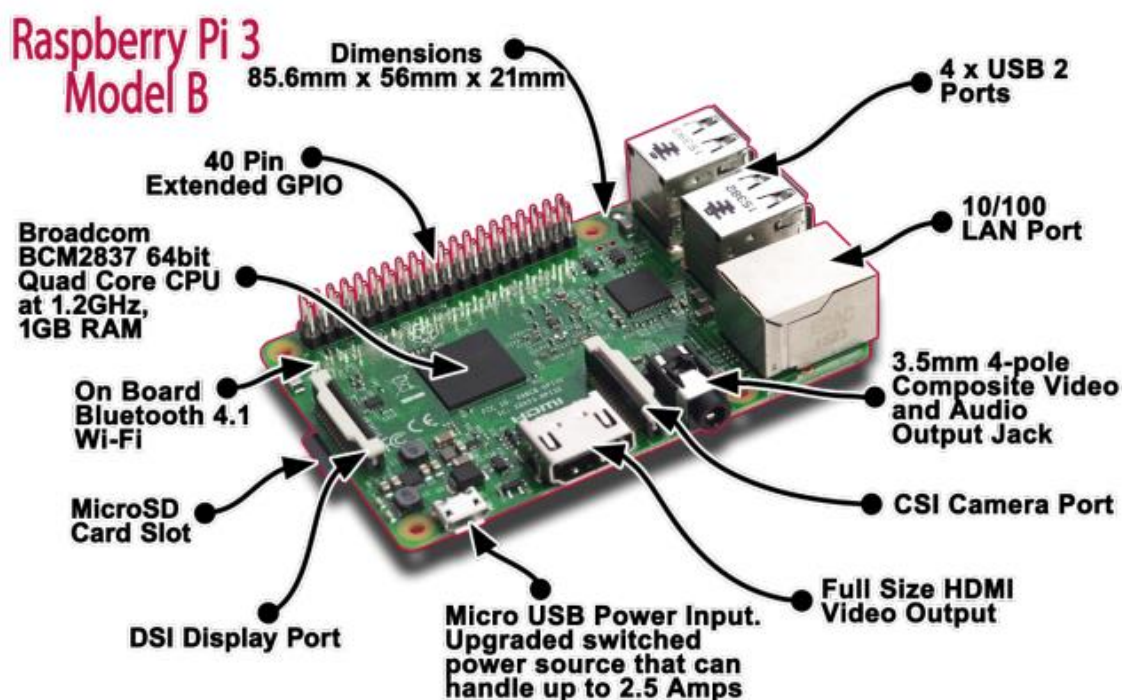


Fig. 9 Conexiones de la Raspberry Pi

Sistema operativo

El sistema operativo instalado y que vamos a utilizar en este proyecto, es *Raspbian*.^[22]

Raspbian es un sistema operativo libre basado en *Debian*^[23] optimizado para el hardware de *Raspberry Pi*. *Raspbian* ofrece más que un SO puro: viene con más de 35.000 paquetes, software pre-compilado empaquetado en un formato que hace más fácil la instalación en su *Raspberry Pi*.

Raspbian es pionero en Sistemas Operativos para *Raspberry*, y cada pocos días aparecen actualizaciones. Su comunidad es muy grande y proactiva, recordando a la de *Debian*, de la cual procede.

Paquetes instalados

Aquí vamos a indicar y a explicar los principales paquetes, los más importantes y necesarios para poder tener nuestro servidor funcional.

MQTT y Mosquitto

Mosquitto es un broker para el protocolo MQTT. Antes de nada explicaremos que es MQTT.

MQTT^[24] (Message Queue Telemetry Transport) es un protocolo de mensajería de publicación-suscripción para mensajes “ligeros”, diseñado para usarse en el último nivel del protocolo TCP/IP. Está diseñado para las conexiones en lugares remotos donde se requiere una pequeña huella en el código y / o el ancho de banda es limitado. El patrón de publicación-suscripción de mensajería requiere un broker de mensajes. El broker es responsable de distribuir mensajes a los clientes interesados basado en el tema del mensaje. Andy Stanford-Clark y Arlen Nipper of Cirrus Link Solutions desarrollaron la primera versión del protocolo en 1999.

La arquitectura de MQTT sigue una topología de estrella, con un nodo central que hace de servidor o broker con una capacidad de hasta 10000 clientes. El broker es el encargado de gestionar la red y de transmitir los mensajes, para mantener activo el canal, los clientes mandan periódicamente un paquete

(PINGREQ) y esperan la respuesta del broker (PINGRESP). La comunicación puede ser cifrada entre otras muchas opciones.

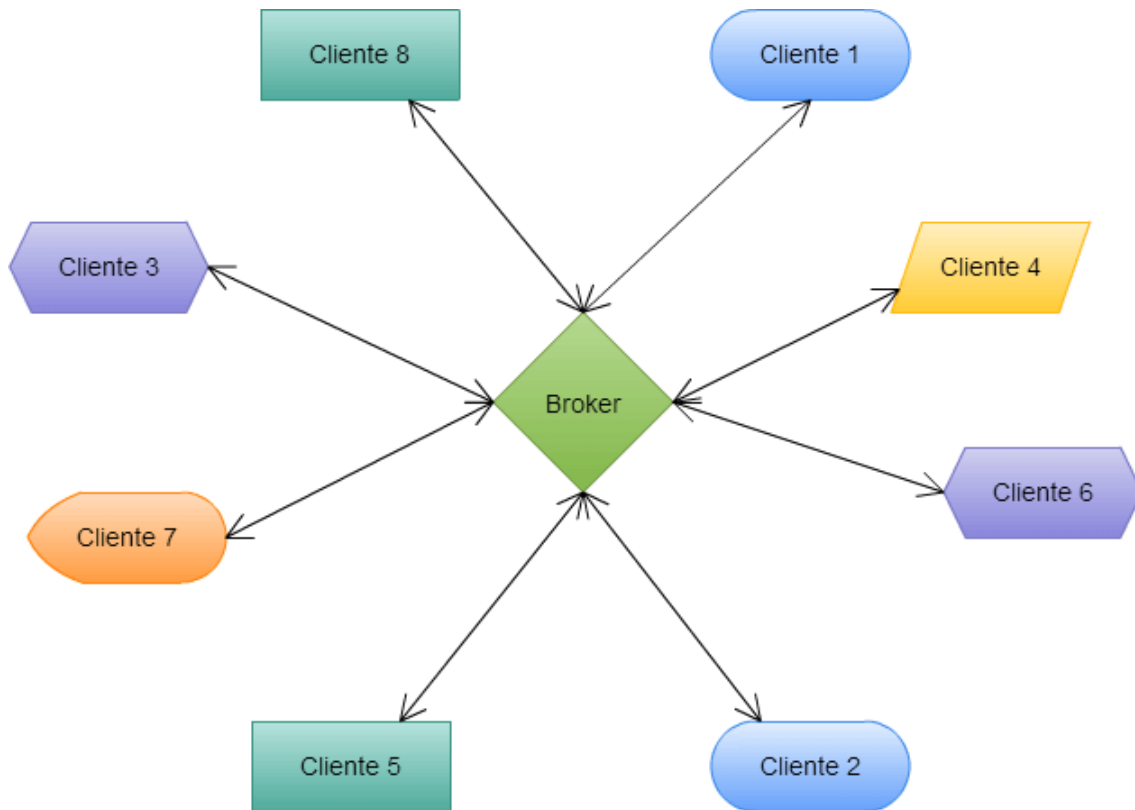


Fig. 10 Conexiones del protocolo Mosquitto

La comunicación se basa en unos “topics” (temas), que el cliente que publica el mensaje crea y los nodos que deseen recibirlo deben subscribirse a él. La comunicación puede ser de uno a uno, o de uno a muchos.

Un “topic” se representa mediante una cadena y tiene una estructura jerárquica. Cada jerarquía se separa con ‘/’. Por ejemplo, “**edificio1/Salon/ModuloWifi/ESP8266**” o “**edificio1/Cocina/ModuloFuego/SensorFuego**”. De esta forma se pueden crear jerarquías de clientes que publican y reciben datos. Un nodo puede subscribirse a un “topic” concreto (“edificio1/PuertaPrincipal/ModuloPuerta/SensorMagnetico”) o a varios (“edificio1/Cocina/#”).

En la siguiente imagen lo podremos ver con más claridad, en ella representamos el mapa jerárquico de nuestro sistema:

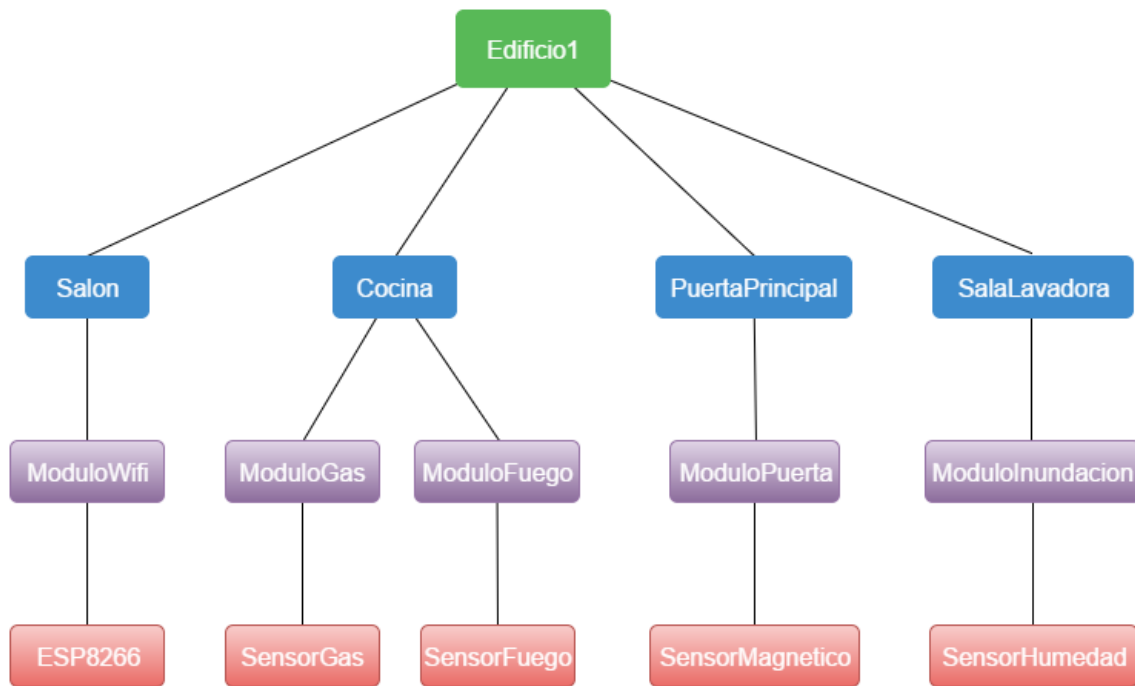


Fig. 11 Mapa de las conexiones MQTT del sistema

Mosquitto ^[25] es un broker de MQTT. Un broker es un servicio que se ejecuta entre el editor y el suscriptor (publisher y suscriber) y proporciona el intercambio de mensajes, QoS (calidad de servicio),...

En nuestro sistema, *Mosquitto* juega una parte fundamental del servidor ya que es el protocolo utilizado para la comunicación entre todos los módulos de *Arduino* y el servidor.

OpenHAB

OpenHAB ^[26] es un software desarrollado en *Java* que pretende integrar múltiples sistemas de domótica que permiten automatizar distintas funciones de la casa, como por ejemplo encender la calefacción o las luces en un determinado momento, o controlar éstas funciones desde tu teléfono móvil. Nosotros lo utilizaremos para poder ver en todo momento el estado de los sensores instalados en la casa del paciente, tanto desde móvil como desde web.

El software puede ejecutarse en cualquier plataforma que soporte *Java 7* o superior. En nuestro caso, lo hemos incluido en una plataforma *Raspberry Pi*, que por su precio y tamaño es ideal para este tipo de soluciones. Además, el proyecto de *openHAB* es opensource y es posible construir soporte para nuevos dispositivos.

OpenHAB tiene múltiples interfaces para controlar los dispositivos que soporta: una aplicación de Android o iOS, una aplicación web o una aplicación de escritorio. OpenHAB nos proporciona una base para éstas aplicaciones, que después se tendrán que configurar, diseñar y adaptar en cada proyecto como convenga. En nuestro caso, hemos elegido la aplicación *Android* y la aplicación *web*, aunque en un futuro también se piensa integrar en la plataforma de *iOS* para dar cobertura a todo tipo de clientes.

Desde un punto de vista de la arquitectura, es una aplicación completamente Java construida sobre el framework OSGI (*Equinox*). Aquí se pueden ver los distintos componentes:

openHAB Architecture Overview

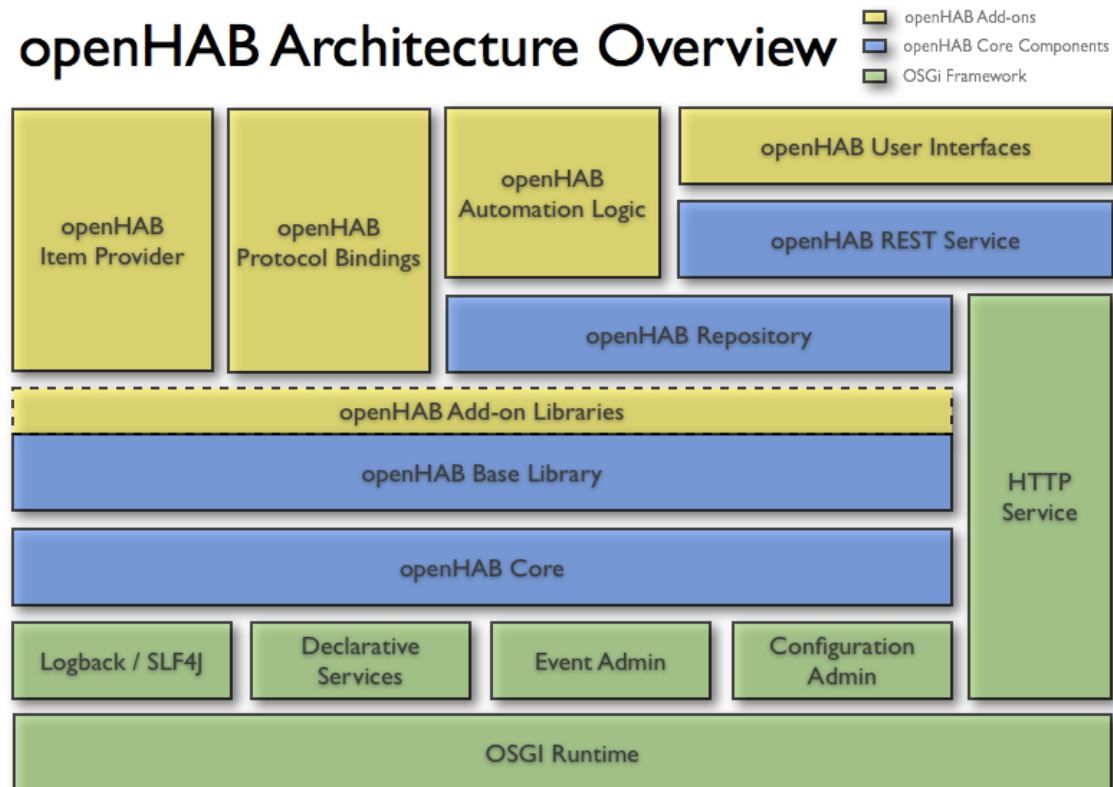


Fig. 12 Arquitectura de openHAB

La comunicación puede realizarse de modo asíncrono a través de un bus de eventos, o puede haber comunicaciones con estado. En este diagrama podemos ver la arquitectura general de las comunicaciones:

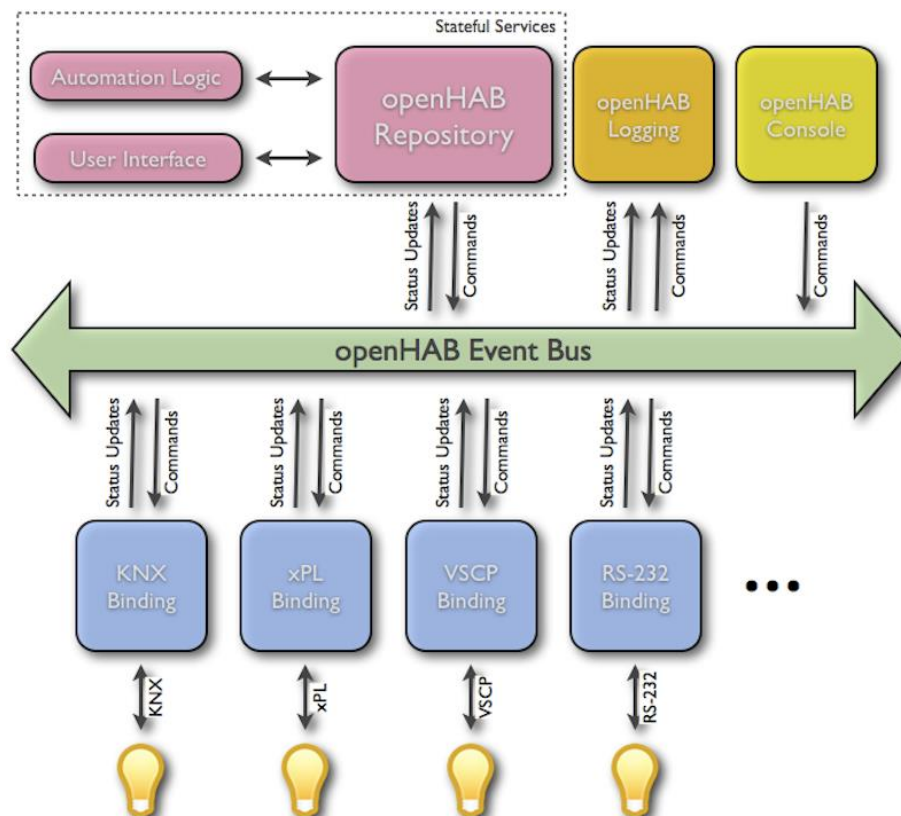


Fig. 13 Comunicaciones de openHAB

My.openHAB

My.openHAB^[40] es un servicio en la nube de código abierto para *openHAB*. Éste servicio proporciona a los usuarios de *openHAB* la capacidad de controlar de forma remota sus casas a través de Internet, recoger y visualizar datos, recibir notificaciones,...

Desde el portal de *my.openHAB* se puede configurar y crear diferentes usuarios, asignarles diferentes roles y administrar qué pueden hacer y que no estos usuarios.

Gracias a éste servicio podremos enviar notificaciones *push* al terminal móvil del usuario para enviarles todos los avisos necesarios que éste necesite o quiera estar informado.

My.openHAB viene preparado para que se puedan utilizar muchas funcionalidades en él, tanto internas como externas, como es el caso de *IFTTT*^[41]. Nosotros no utilizaremos esta herramienta ya que trabajamos internamente con *openHAB*, pero *IFTTT* es una herramienta muy conocida y muy útil y es un apunte muy interesante que tiene *my.openHAB*.

12. DISEÑO DE LOS MÓDULOS ARDUINO

En este proyecto utilizaremos la plataforma *Arduino*, en la que nos apoyaremos con otros dispositivos para poder realizar nuestro proyecto.

Los actuadores no han sido aún implementados, y por lo tanto, estarán especificados pero el diseño no se incluirá en esta documentación.

Que es Arduino?

Arduino^[27] es una plataforma de hardware libre creada en 2005, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios. Está especialmente indicada para artistas, diseñadores, aficionados y para cualquier persona interesada en crear objetos o espacios interactivos.

Al ser *Arduino* una plataforma de hardware libre, tanto su diseño como su distribución puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia. En la placa *Arduino* es donde conectaremos los sensores, actuadores y otros elementos necesarios para comunicarnos con el sistema. En el proyecto se han utilizado las placas *Arduino UNO R3*.

Arduino UNO R3

Arduino UNO R3^[28] es una placa con un microcontrolador de la marca *Atmel* y con toda la circuitería de soporte, que incluye, reguladores de tensión, un puerto USB conectado a un módulo adaptador USB-Serie que permite programar el microcontrolador desde cualquier PC y también hacer pruebas de comunicación con el propio chip.

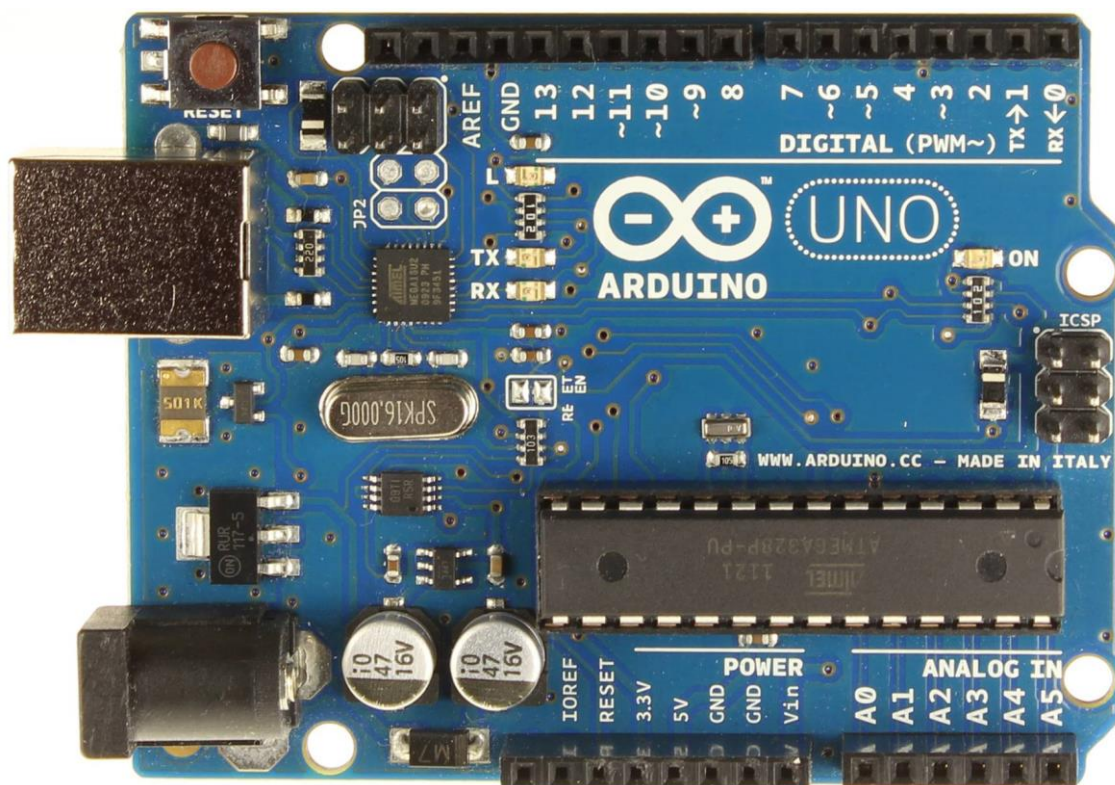


Fig. 14 Arduino UNO R3

Características técnicas

- Microcontrolador: ATmega328.
- Voltaje de operación: 5V.
- Voltaje de entrada (recomendado): 7-12V.
- Voltaje de entrada (límite): 6-20V.
- 14 pines digitales de I/O (6 salidas PWM).
- 6 pines analógicos.
- Corriente continua por pin IO: 40 mA.
- Corriente continua por el pin 3.3V: 50 mA.
- 32k de memoria Flash (0,5 KB ocupados por el *bootloader*).
- SRAM de 2KB.
- EEPROM de 1KB
- Reloj de 16MHz de velocidad.

Conexiones y pines

Una placa *Arduino* dispone de 14 pines que pueden configurarse como entrada o salida y a los que puede conectarse cualquier dispositivo que sea capaz de transmitir o recibir señales digitales de 0 y 5 V. También dispone de entradas y salidas analógicas. Mediante las entradas analógicas podemos obtener datos de sensores en forma de variaciones continuas de un voltaje. Las salidas analógicas suelen utilizarse para enviar señales de control en forma de señales PWM.

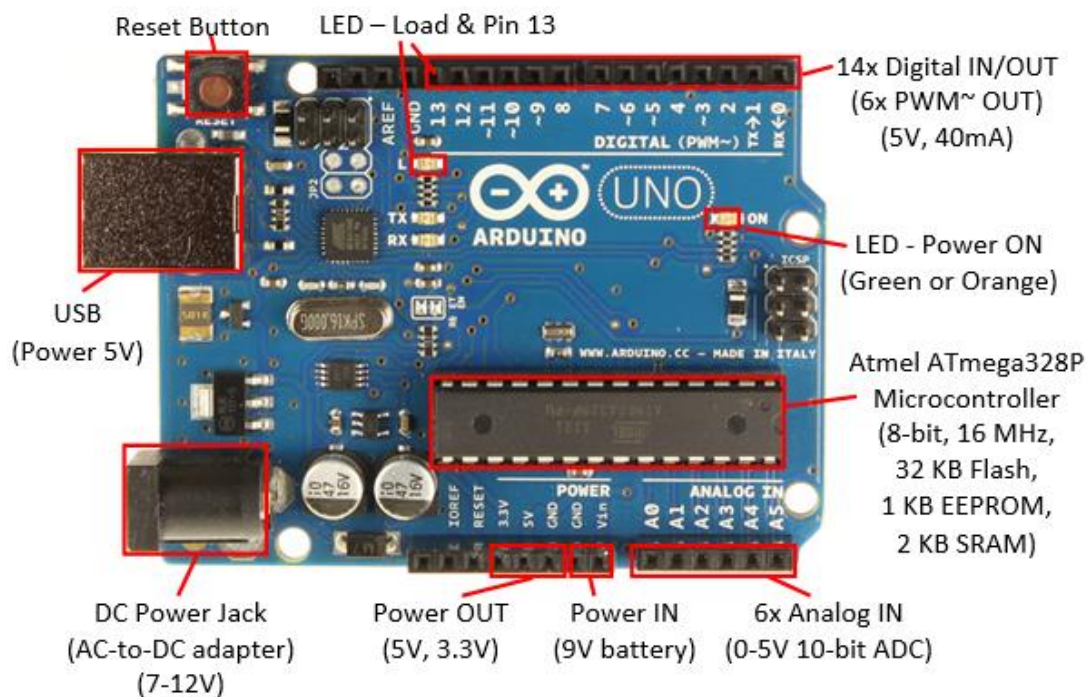


Fig. 15 Conexiones y pines de la placa Arduino UNO R3

Módulos de Arduino

Para implementar todas las funcionalidades que requiere el sistema, se necesitan diferentes módulos de *Arduino* con unos componentes específicos. Cada módulo irá colocado estratégicamente en diferentes habitaciones de la casa para cumplir su funcionalidad.

Los módulos irán conectados siguiendo el modelo de una conexión en estrella. El módulo central será el módulo Wif, que será el encargado de comunicarse con todos los otros módulos. Éste módulo será el encargado de tratar toda la información recibida de los módulos y enviársela al servidor. La conexión con el servidor será mediante WiFi, pero la conexión con todos los otros módulos *Arduino* se hará mediante *ZigBee*. Antes de entrar en detalle en cada módulo, vamos a explicar en qué consiste el *ZigBee*, ya que será una herramienta compartida en todos los módulos.

ZigBee

ZigBee ^[29] es un estándar abierto para radiocomunicaciones de baja potencia basado en la especificación 802.15.4 ^[30] de redes inalámbricas de área personal o WPAN ^[31]. Incluye un robusto y confiable protocolo de red, de bajo consumo y coste, con servicios de seguridad y capa de aplicación que garantiza la interoperabilidad entre dispositivos. *ZigBee* trabaja a una velocidad de datos de 250 Kbps sobre la banda libre de 2,4 Ghz, aunque también soporta las bandas de 868 y 900 MHz.

El estándar *ZigBee* es publicado y administrado por un grupo de empresas y fabricantes denominado *ZigBee Alliance* ^[32]. Por tanto, el término *ZigBee* es en realidad una marca registrada de este grupo y no un estándar técnico. Sin embargo, para fines no comerciales, la especificación *ZigBee* se encuentra disponible de forma abierta y libre al público. De esta forma la relación entre IEEE 802.15.4 y *ZigBee Alliance* es similar a aquella entre IEEE 802.11 ^[33] y Wi-Fi Alliance ^[34].

Principales características de *ZigBee*:

- Bajo coste: estándar abierto que hace innecesario el pago de patentes.
- Bajo consumo energético: permite prolongar ampliamente la vida de las baterías.
- Sencillez: cuenta con una pila de protocolos de reducido tamaño.
- Alta confiabilidad: redes malladas con redundancia de enlaces y canales y posibilidad de enrutamiento alternativo automático.
- Despliegue de red sencillo: fácil de montar y administrar gracias a las redes ad-hoc y al enrutamiento automático.
- Alta seguridad: proporciona integridad de datos y autenticación, haciendo uso del algoritmo de cifrado AES-128.
- Compatibilidad: al tratarse de un estándar abierto se garantiza la interoperabilidad entre dispositivos de diferentes fabricantes.
- Baja latencia: los retardos producidos en las distintas tareas de red son muy reducidos, del orden de milisegundos frente a varios segundos en otras tecnologías.
- Gran capacidad: una misma red puede soportar más de 65000 nodos independientes.

XBee

XBee ^[35] es el hardware *ZigBee* utilizado en el proyecto. Se trata de una gama de módulos de radio fabricados por *Digi International* ^[36] con una amplia variedad de modelos, componentes, firmware, potencias de transmisión y antenas. Trabajan con un voltaje de operación de 3,3 voltios y, aunque incorporan hasta 20 pines para distintas operaciones como reinicio o modo de bajo consumo, pueden funcionar haciendo uso tan sólo de sus 4 pines principales: alimentación, tierra, y entrada y salida de datos.

Cuentan con un microchip que, además de ocuparse de las tareas propias de la comunicación radio, les permite realizar operaciones con un nivel de lógica muy básico, como por ejemplo leer directamente datos de sensores y retransmitirlos sin tener que contar con procesado externo.

El módulo *XBee* utilizado en este proyecto es el Serie 2. Estos módulos permiten implementar redes malladas siguiendo el protocolo *ZigBee*. Además poseen un mayor alcance y un menor consumo que la serie 1. Estas características convierten a los módulos de la serie 2 en los idóneos para implementar una red de sensores como la que ocupa este proyecto.

La placa *Arduino* va conectada con el módulo *XBee* mediante una *ZigBee Shield* ^[37]. Se trata de un adaptador especialmente diseñado para conectar una placa *Arduino* y un módulo *XBee*. Realiza todas las conexiones necesarias entre los pines de ambos elementos y dota de estabilidad al conjunto. Proporciona fácil acceso a los conectores libres de la placa, junto con un botón adicional de reinicio del microcontrolador. Dispone además de un interruptor que permite seleccionar la salida serie del módulo *XBee* entre radio o USB, lo que facilita su configuración.

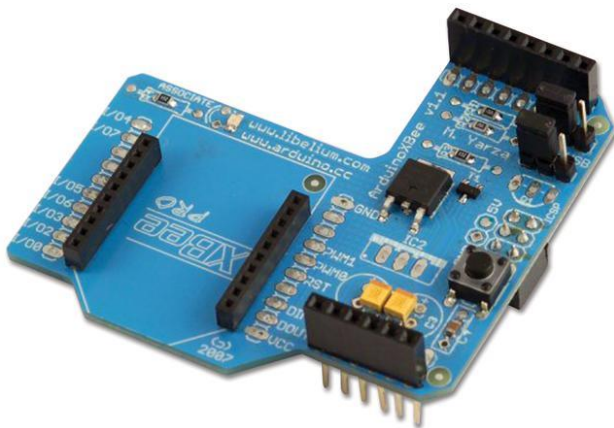


Fig. 16 ZigBee shield

La *ZigBee shield* se conecta con la placa *Arduino* directamente encima de ésta. A continuación mostraremos una imagen donde se puede ver cómo queda la placa *Arduino* con la *ZigBee Shield* montada encima de él.

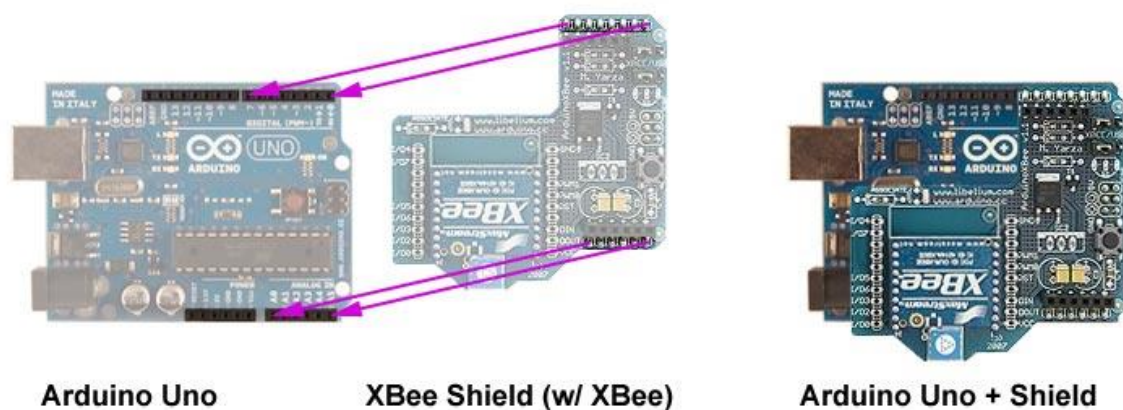


Fig. 17 Montaje de la ZigBee shield en la placa Arduino

Y por último, ya solo nos falta el módulo *Xbee Serie 2*. Es el encargado de la comunicación inalámbrica con los demás nodos e implementa el protocolo *ZigBee*, que permite la conexión y reconfiguración automática de la red. Dependiendo de su configuración actuará como coordinador, router o terminal. Éste módulo también va conectado encima de la *ZigBee shield*, encajando perfectamente con todos los pines necesarios conectados.



Fig. 18 Módulo Xbee Serie 2

Módulo Wifi

El módulo Wifi es el módulo central de los *Arduino*. Su colocación no es importante, ya que se comunica inalámbricamente con los otros componentes y no incorpora ningún sensor que necesite una ubicación determinada.

Actúa como coordinador de la red, siendo él quien se comunica con todos los otros módulos, recibiendo los datos que emiten y, en caso necesario, enviándoles los mensajes necesarios. Ésta comunicación se hace a través de *XBee*, como ya hemos comentado en el punto anterior.

Éste módulo es también el que se comunica con la *Raspberry Pi* (servidor), para transmitirle todos los datos. Ésta comunicación se hace a través del protocolo *Mosquitto*, también explicado ya en un punto anterior. Pero, cómo un *Arduino* puede comunicarse con una *Raspberry Pi* a través de WiFi? La respuesta es muy fácil: a través del módulo *ESP8266* ^[38].

Módulo ESP8266

El *Módulo Wifi Serial ESP8266* ofrece una solución completa y muy económica para conexión de sistemas a redes Wi-Fi. El *módulo wifi serial ESP8266* es capaz de funcionar como “adaptador de red” en sistemas basados en microcontroladores que se comunican con él a través de una interfaz UART. Es así como lo hemos utilizado nosotros, lo hemos flasheado y le hemos cargado un código para que funcione como de intermediario entre la comunicación entre el módulo Wifi de *Arduino* y el servidor de *Raspberry Pi*. Se comunica con el conocido protocolo de *MQTT*.

Se trata posiblemente del módulo Wifi serial más económico en el mercado y que además concentra las características mínimas necesarias para poder utilizarse sin problemas, ya que hay una gran comunidad alrededor de él. Se usa como una alternativa barata y confiable para habilitar los productos y prototipos para la era del Internet de las cosas (IoT).



Fig. 19 Módulo ESP8266

Características del módulo Wifi Serial ESP8266:

- Protocolos soportados: 802.11 b/g/n
- Wi-Fi Direct (P2p), Soft Access Point
- Stack TCP/IP integrado
- PLL, reguladores y unidades de manejo de energía integrados
- Potencia de salida: +19.5dBm en modo 802.11b
- Consumo en modo de baja energía: <10 uA
- Procesador integrado de 32 bits, puede ser utilizado como procesador de aplicaciones

Para poder flashear éste módulo y así convertirlo en el comunicador entre los módulos *Arduino* y el servidor, se utilizó una placa UART ^[39].

La UART, acrónimo de *Universal Asynchronous Receiver-Transmitter*, es un chip cuya misión principal es convertir los datos recibidos del bus del PC en formato paralelo, a un formato serie que será utilizado

en la transmisión hacia el exterior. También realiza el proceso contrario: transformar los datos serie recibidos del exterior en un formato paralelo entendible por el bus.

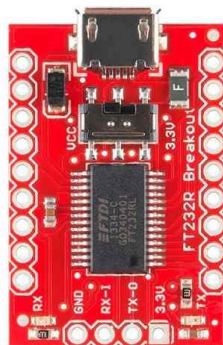


Fig. 20 UART

Interfaces

Leds

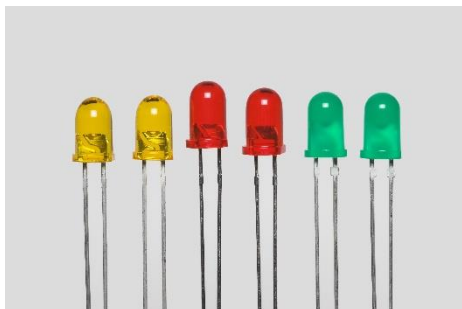


Fig. 21 Leds

La función de este dispositivo es avisar mediante un diodo emisor de luz la ocurrencia de un peligro que pueda ocurrir en la casa, y así avisar a la persona responsable de éste.

Timbre

Este elemento es capaz de producir sonidos. Es utilizado para avisar al paciente sobre un problema grave en la vivienda dado que la sensibilidad auditiva es mayor que la visual.



Fig. 22 Zumbador

Pulsador



Para que el paciente pueda comunicarse con el sistema necesitaremos un pulsador (como un botón) como éste. La función de presionar el botón servirá para apagar la alarma y comunicarle así al sistema que el paciente está al tanto del peligro y está poniendo una solución.

Fig. 23 Pulsador

Circuito

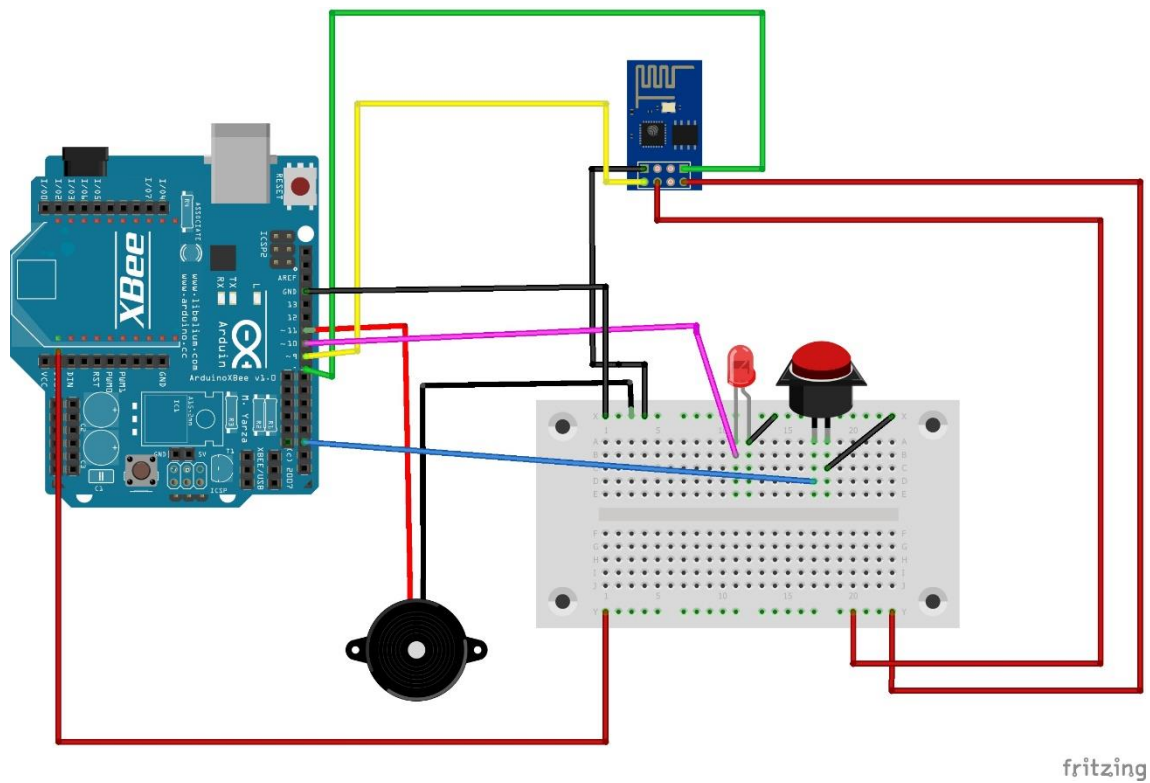


Fig. 24 Circuito del módulo Wifi

Módulo de Gas

Éste es el módulo que detecta las posibles fugas de gas que habrá. Se ubicará en la cocina.

Sensores

Gas



El detector de gas detecta la cantidad de gas que hay en el ambiente. Cuando detecta que la cantidad de gas en el ambiente es superior a un mínimo (posible por tener los fogones encendidos), envía una alerta de que hay poco gas. Cuando supera unos máximos, la alerta cambia a mucho gas.

Fig. 25 Sensor de Gas

Circuito

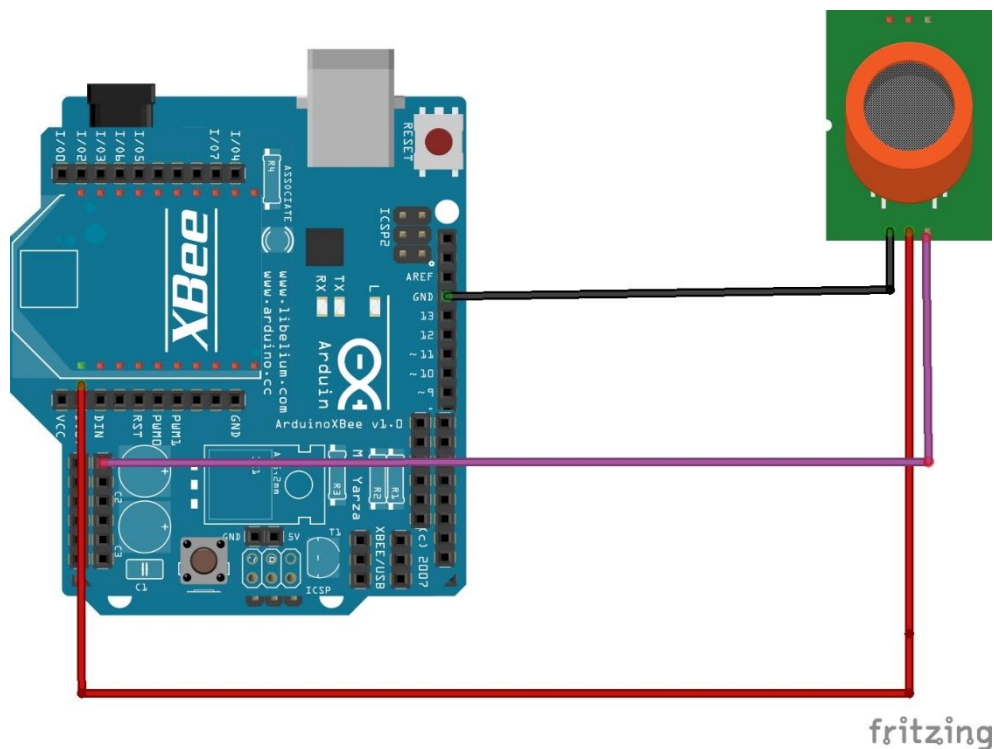


Fig. 26 Circuito del módulo de Gas

Módulo de Fuego

Éste es el módulo que detecta los posibles incendios que pueda haber. Se ubicará en la cocina.

Sensores

Fuego



El detector de llama detecta, mediante infrarrojos, la distancia que hay hasta un foco de fuego. Cuando detecta que la distancia de un foco de fuego es superior a un mínimo (posible por tener los fogones encendidos), envía una alerta de que hay fuego lejos. Cuando supera unos máximos, la alerta cambia a fuego cerca.

Fig. 27 Sensor de llama

Circuito

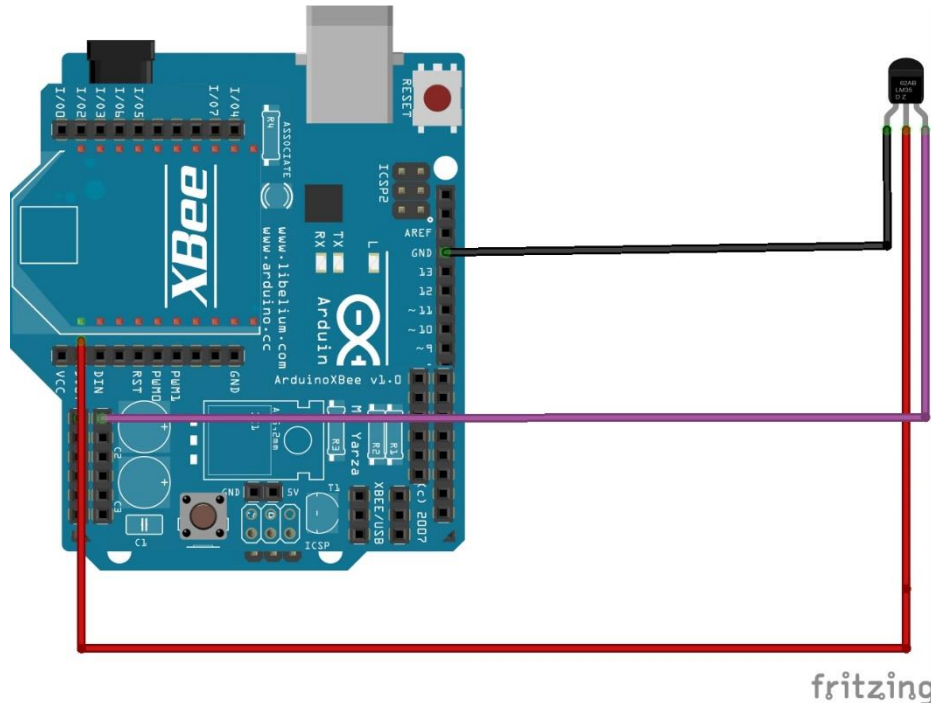


Fig. 28 Circuito del módulo de fuego

Módulo de Puerta Abierta

Éste es el módulo que detecta cuando un paciente se ha dejado la puerta abierta demasiado tiempo. Se ubicará en la puerta principal.

Sensores

Magnético



El sensor magnético detecta cuando una puerta está abierta y cuando está cerrada. Cuando detecta que la puerta está abierta, acciona un contador para contar el tiempo que está abierta. Cuando pasa cierto tiempo y la puerta sigue abierta, se lanza una alerta para indicar que la puerta lleva demasiado tiempo abierta.

Fig. 29 Sensor magnético

Circuito

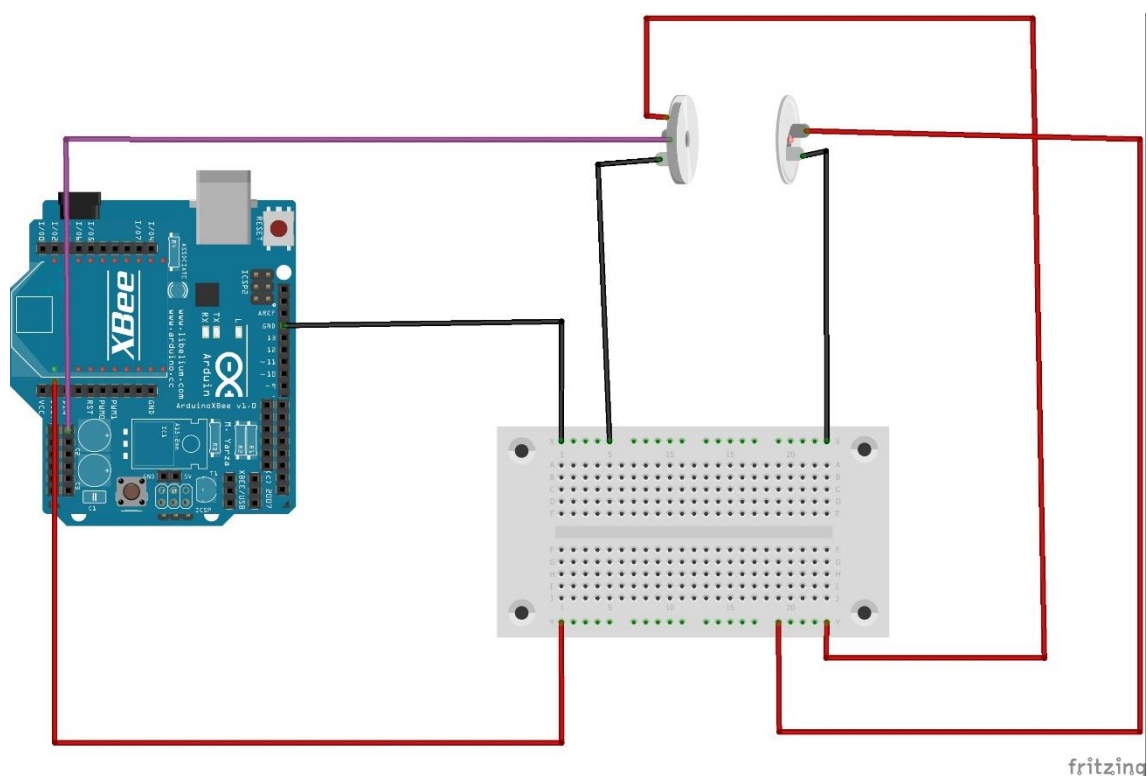


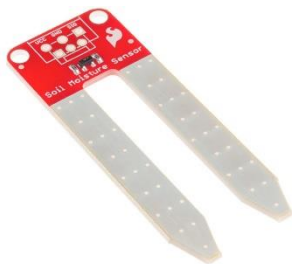
Fig. 30 Circuito del módulo de Puerta Abierta

Módulo de Inundación

Éste es el módulo que detecta cuando hay una inundación, ya sea por una fuga de agua de algún electrodoméstico como por cualquier otra causa. Se ubicará en la sala de la lavadora.

Sensores

Humedad



El sensor de humedad detecta la cantidad de agua que hay en el terreno. Cuando detecta que la cantidad de agua que hay en el terreno es superior a un mínimo (posible al fregar o similares), envía una alerta de que hay una posible inundación.

Fig. 31 Sensor de humedad

Circuito

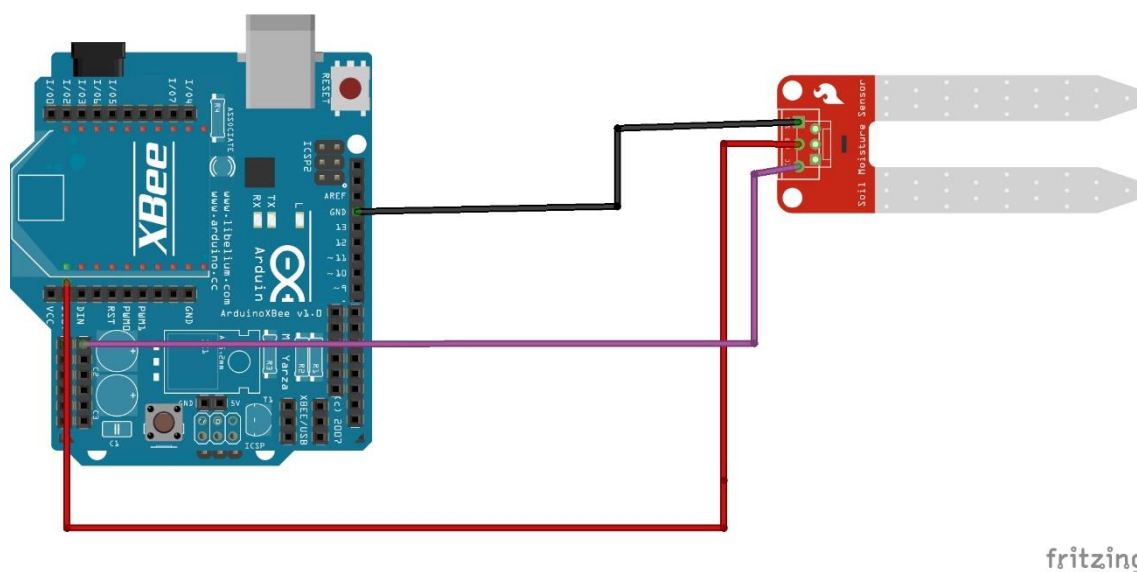


Fig. 32 Circuito del módulo de Inundación

13. APLICACIÓN WEB

La aplicación web de *openHAB* viene diseñada por su propia herramienta por lo tanto no describiremos su arquitectura, aunque si vamos a mostrar las pantallas que tendrá visibles el usuario y la interacción que tendrá con ella, que serán las pantallas diseñadas e implementadas en el servidor.

Para acceder a la aplicación web remotamente, hay que conectarse a la url:

<https://my.openhab.org/openhab.app?sitemap=ConHome>

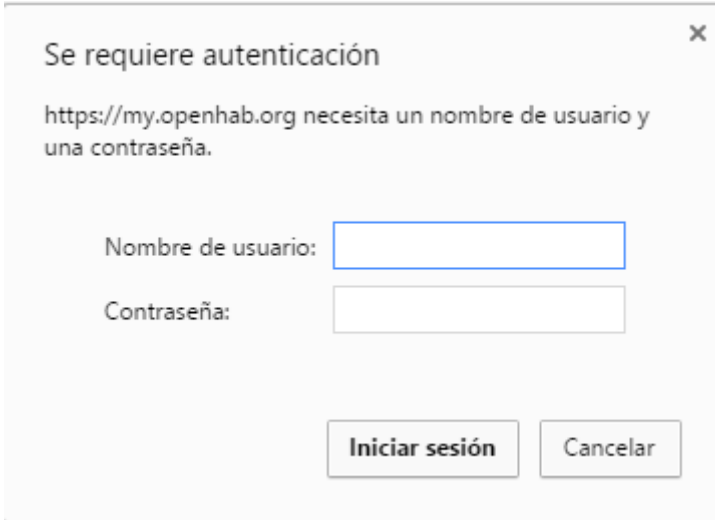
Cuando se intente acceder a la web, el sistema pedirá al usuario que se identifique con un usuario y una contraseña, que se le será facilitada previamente.

13.1 Diseño de las pantallas

A continuación se verá como se han diseñado las pantallas y como cambian o actúan después de cada interacción.

Pantalla de login

En la pantalla de login, si el usuario es correcto le envía a la página principal, sino le aparece un mensaje de login incorrecto.



A screenshot of a login dialog box titled "Se requiere autenticación" (Authentication required). The dialog box has a close button (X) in the top right corner. Below the title, it says "https://my.openhab.org necesita un nombre de usuario y una contraseña." (https://my.openhab.org needs a username and a password). There are two input fields: "Nombre de usuario:" (Username) and "Contraseña:" (Password). At the bottom, there are two buttons: "Iniciar sesión" (Log in) and "Cancelar" (Cancel).

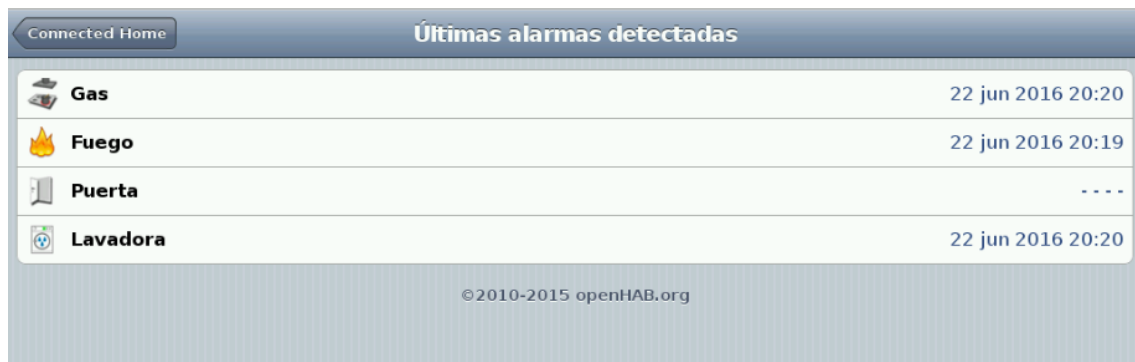
Pantalla principal

Esta es la pantalla principal, donde se muestran todos los sensores que el cliente haya contratado y que se hayan configurado e instalado en la casa del paciente.



Pantalla de alarmas

Esta es la pantalla donde se muestra la hora y la fecha de las últimas alarmas que se han detectado para cada uno de los sensores o módulos instalados.



13.2 Interacción con el usuario

La pantalla principal cambia de estado según la detección de los diferentes peligros. A continuación se muestra como se muestra la pantalla con cada cambio o peligro detectado.

Poco gas



Mucho gas



Fuego lejos

Connected Home

Cocina


Gas detectado

SIN GAS


Fuego detectado

FUEGO LEJOS

Puerta Principal


Puerta abierta

PUERTA CERRADA

Sensores de Inundación


Fuga lavadora

SECO

Últimas Alarmas


Últimas alarmas detectadas

>

©2010-2015 openHAB.org

Fuego cerca

Connected Home

Cocina


Gas detectado

SIN GAS


Fuego detectado

FUEGO CERCA

Puerta Principal


Puerta abierta

PUERTA CERRADA

Sensores de Inundación


Fuga lavadora

SECO

Últimas Alarmas


Últimas alarmas detectadas

>

©2010-2015 openHAB.org

Puerta abierta

Connected Home

Cocina

 **Gas detectado** SIN GAS

 **Fuego detectado** SIN FUEGO

Puerta Principal

 **Puerta abierta** PUERTA ABIERTA

Sensores de Inundación

 **Fuga lavadora** SECO

Últimas Alarmas

 **Últimas alarmas detectadas** >

©2010-2015 openHAB.org

Inundación

Connected Home

Cocina

 **Gas detectado** SIN GAS

 **Fuego detectado** SIN FUEGO

Puerta Principal

 **Puerta abierta** PUERTA CERRADA

Sensores de Inundación

 **Fuga lavadora** INUNDADO

Últimas Alarmas

 **Últimas alarmas detectadas** >

©2010-2015 openHAB.org

14. APLICACIÓN MÓVIL

Aunque por supuesto se puede utilizar la interfaz web desde un dispositivo móvil, *openHAB* tiene aplicaciones nativas para *iOS* y *Android* – y tienen un diseño y una usabilidad mucho mejor que la interfaz del navegador por defecto. Nosotros, solo hemos implementado y probado la aplicación *Android*, por lo tanto solo aseguramos que sea esa la que funciona correctamente, aunque si se prueba desde un terminal *iOS* seguramente funcionarán todas o casi todas las funcionalidades del sistema.

Para acceder remotamente desde la aplicación móvil al sistema, es muy fácil. Simplemente hay que rellenar, en la pantalla de configuración, la URL remota con:

<https://my.openhab.org>

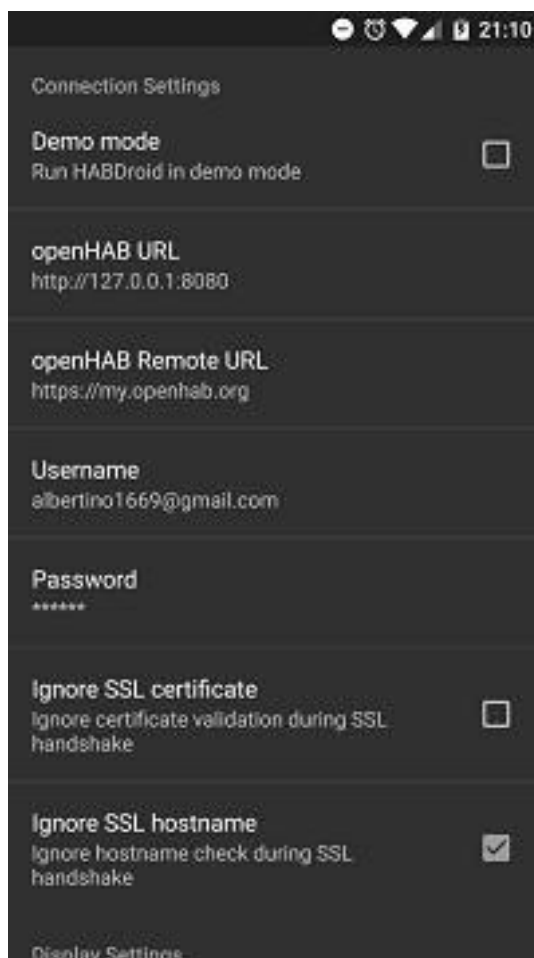
Y después introducir el usuario y la contraseña que se le habrá facilitado previamente al usuario.

14.1 Diseño de las pantallas

A continuación se verá como se han diseñado las pantallas y como cambian o actúan después de cada interacción.

Pantalla de configuración y login

En la pantalla de configuración es donde el usuario tiene que configurar su aplicación con unos parámetros concretos para conectarse a su vivienda. Esta configuración se le proporcionará al usuario para que solo tenga que pasarla de las instrucciones (que se le hará llegar por carta o por correo) a su terminal móvil. Si el usuario es correcto le envía a la página principal, sino le aparece un mensaje de login incorrecto.



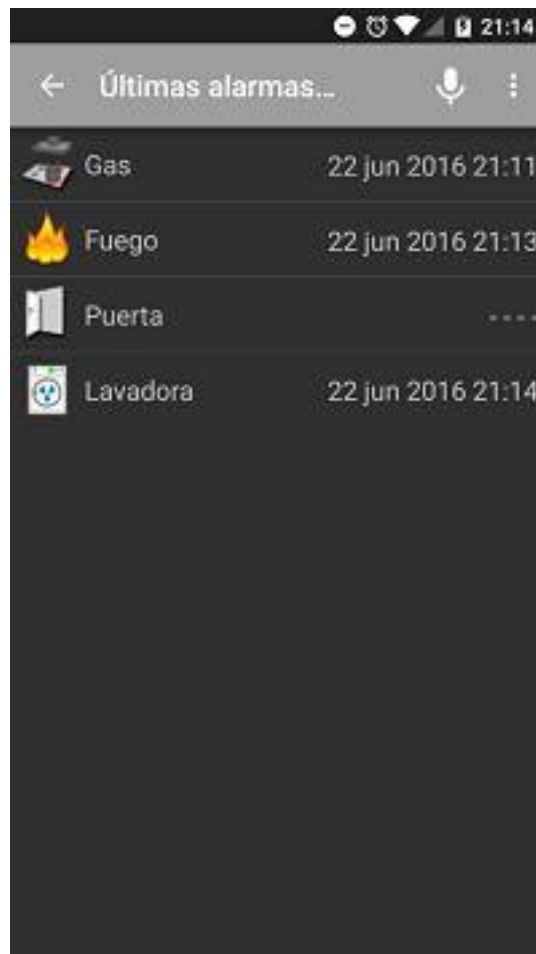
Pantalla principal

Esta es la pantalla principal, donde se muestran todos los sensores que el cliente haya contratado y que se hayan configurado e instalado en la casa del paciente.



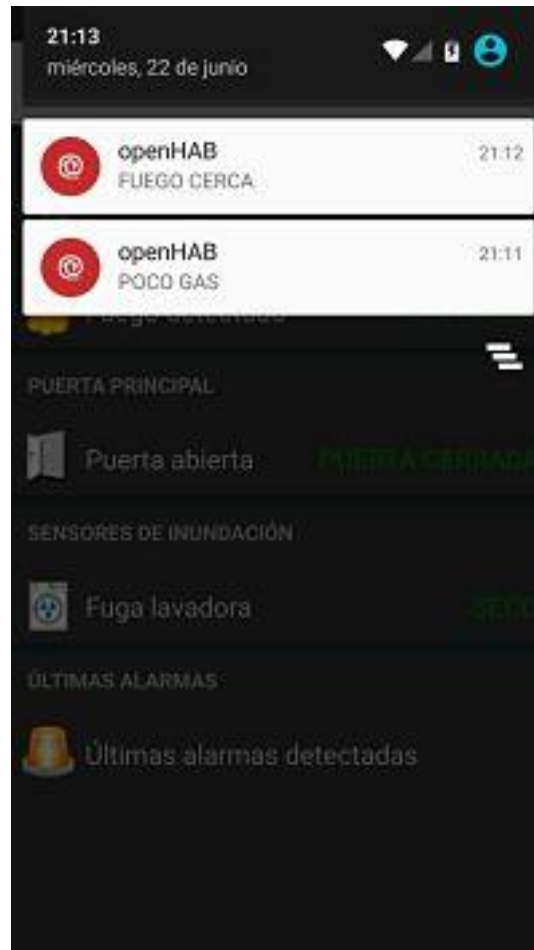
Pantalla de alarmas

Esta es la pantalla donde se muestra la hora y la fecha de las últimas alarmas que se han detectado para cada uno de los sensores o módulos instalados.



Notificaciones

A continuación se muestra el estilo que seguirán las notificaciones push que se le enviarán al usuario a través de su teléfono móvil. Mostramos un ejemplo con dos tipos de notificación, pero todas siguen el mismo criterio.



14.2 Interacción con el usuario

La pantalla principal cambia de estado según la detección de los diferentes peligros. A continuación se muestra como se muestra la pantalla con cada cambio o peligro detectado.

Poco gas



Mucho gas



Fuego lejos



Fuego cerca



Puerta abierta



Inundación



15. INTERACCIÓN ENTRE COMPONENTES DEL SISTEMA

A continuación veremos cómo se relacionan los componentes del sistema. Primero veremos el diagrama de componentes, para entender que forma cada parte del sistema y con quien se comunica. Después veremos el diagrama de conexiones, para ver de qué forma se comunican todos los componentes. Y para finalizar veremos el diagrama de software, para ver cómo está implementado cada componente del sistema.

Diagrama de componentes

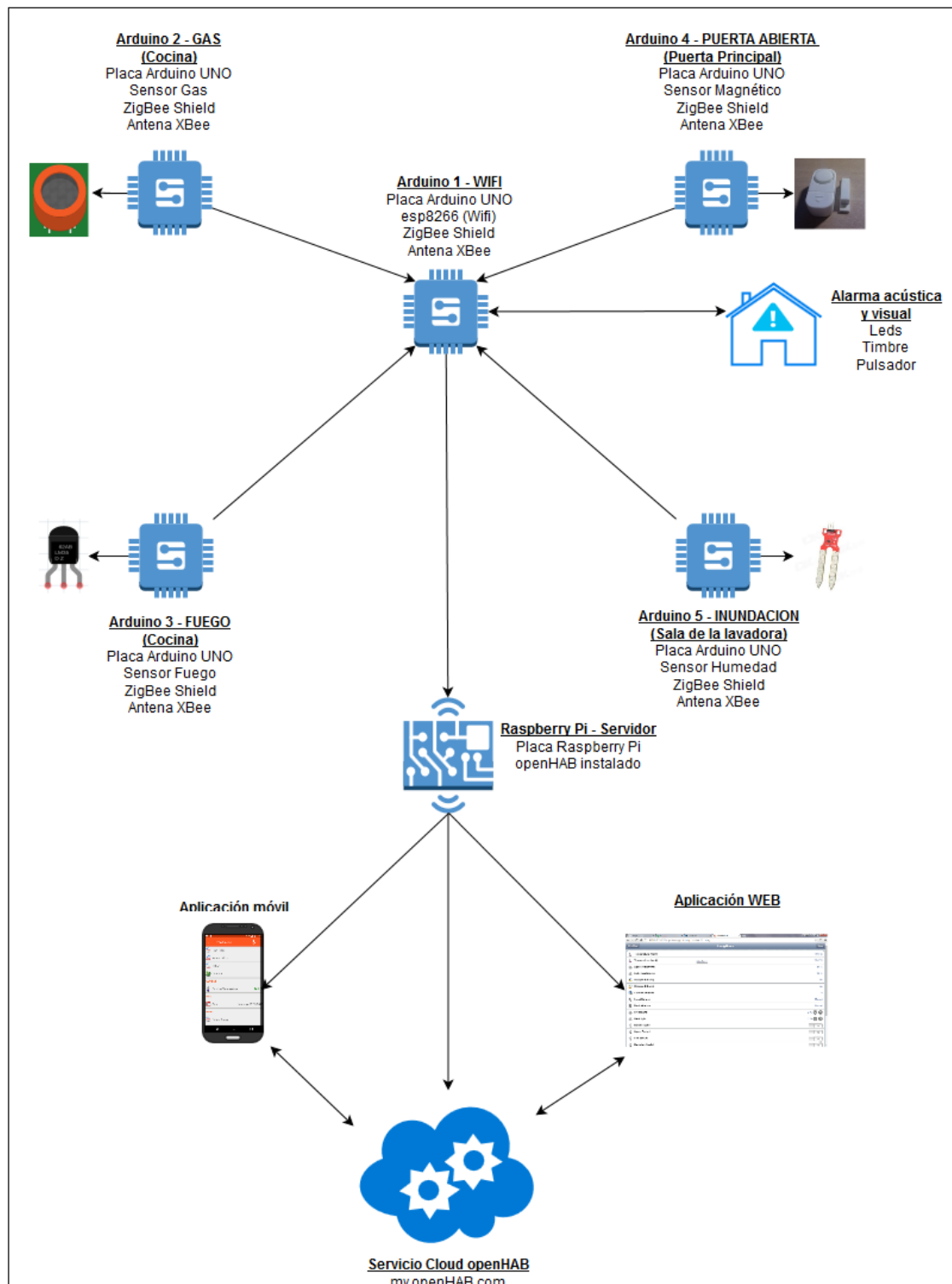


Fig. 33 Diagrama de componentes

Diagrama de conexiones

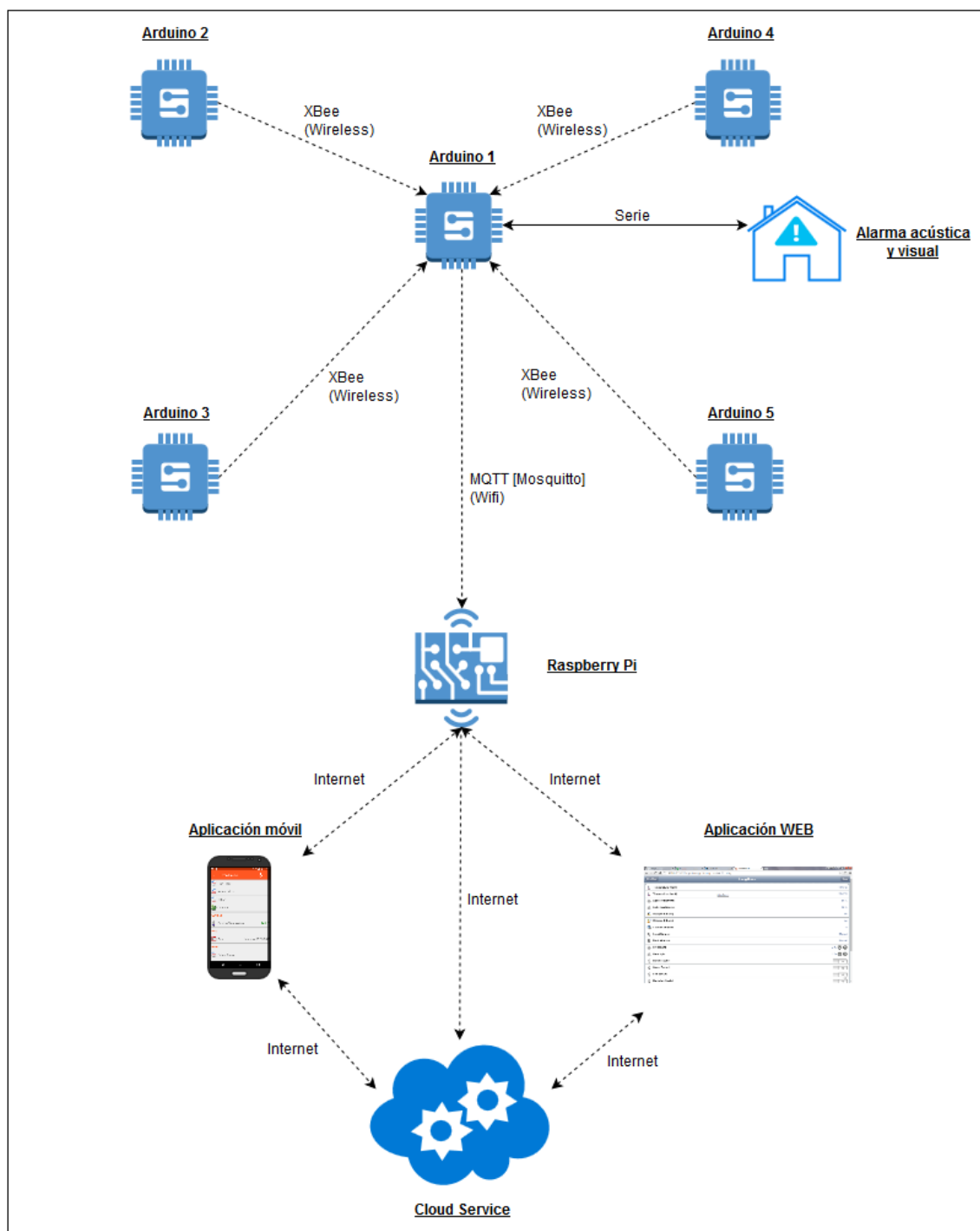


Fig. 34 Diagrama de conexiones

Diagrama de software

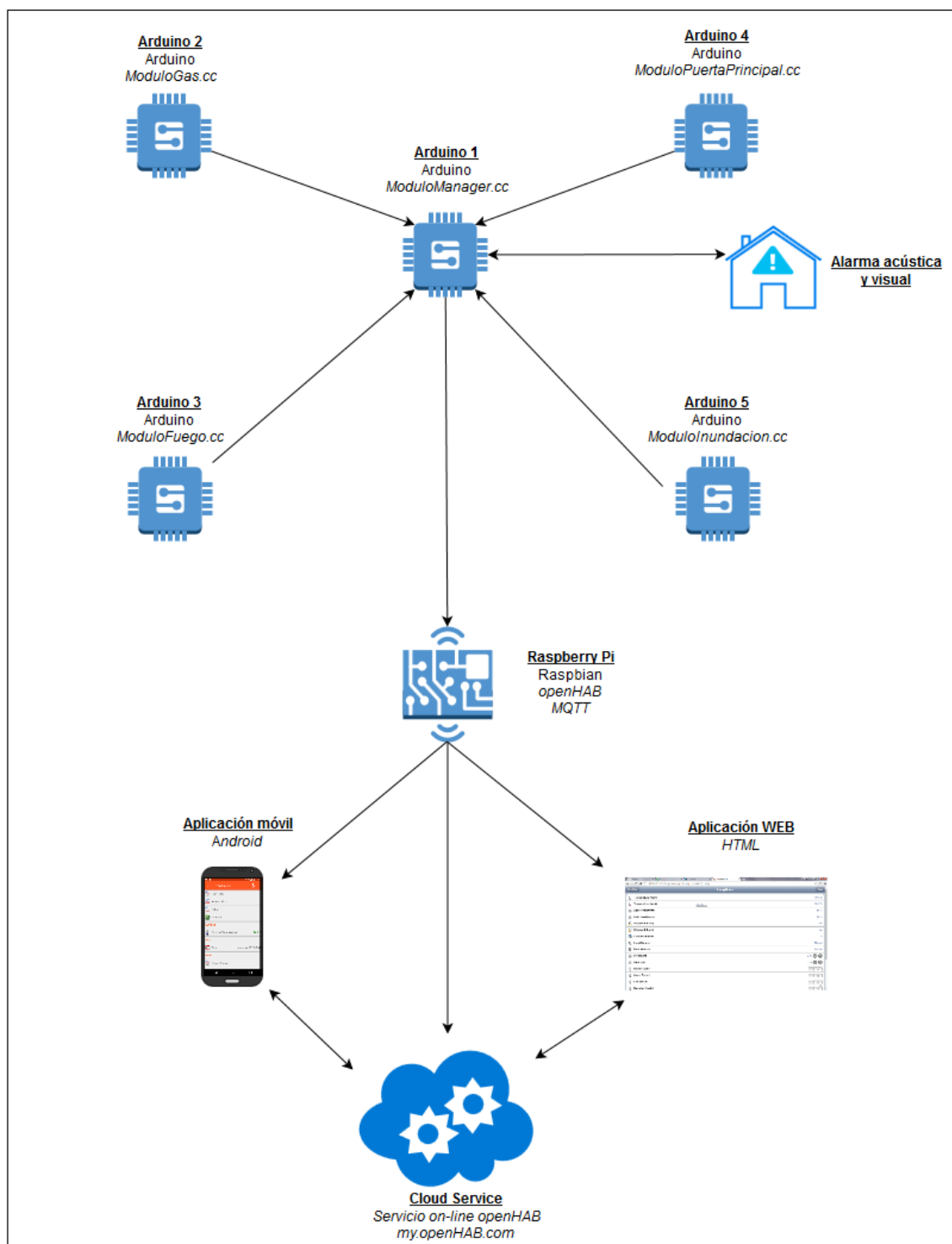


Fig. 35 Diagrama de software

16. CONCLUSIONES

Este proyecto es la elaboración de un sistema compuesto por tres partes: los módulos *Arduino*, el servidor en *Raspberry Pi* y la interacción con usuario mediante las plataformas web y móvil.

Es importante mencionar todos los conocimientos que he adquirido desarrollando este proyecto.

En primer lugar, con los módulos *Arduino*, me he dado cuenta de la verdadera potencia que tienen unos microcontroladores. Son limitados, claro, pero unas placas tan pequeñas y al alcance de todo el mundo pueden hacer cosas muy grandes y muy útiles. También me he dado cuenta de la importancia de la comunidad en este mundo, ya que te puedes encontrar errores de todo tipo, desde el más tonto hasta el más grave. Yo, por ejemplo, he estado como 2 o 3 meses configurando el sistema teniendo un problema que no sabía cómo abordar: se me reiniciaban los módulos *Arduino* cuando tenían la *ZigBee shield* con el módulo *Xbee* conectado. Pregunté por foros, a conocidos, expertos, busqué información, probé mil cosas y nada funcionaba. Desde que el puerto USB no me daba suficiente corriente para alimentarlo hasta que la placa no podía procesar toda la información a la vez, incluso pensé que era un problema de la EEPROM. Finalmente, era tan fácil como que el módulo *XBee* tenía un pin (el D7, no me olvidaré en la vida) que había que ponerlo a 0, sino el *XBee* entraba en modo “sleep” y la placa se reiniciaba.

En cuanto al servidor, me he divertido mucho y aprendido aún más montando esta parte. Había oído hablar de las *Raspberry Pi* pero nunca había visto ni tocado una. Y el protocolo *MQTT* también me ha parecido muy interesante y muy potente. Pero lo que realmente me ha impresionado, es *openHAB*. Una herramienta muy potente y que ofrece un montón de *bundles* para poder hacer y llevar a cabo cualquier tipo de proyecto de domótica (que yo pienso que es el futuro). Me ha impresionado esta herramienta, y estoy seguro de que la volveré a utilizar en un futuro, no solo a nivel profesional, sino a nivel personal, haciendo alguna pequeña domotización en mi propio hogar con estas herramientas.

Finalmente, las plataformas web y móvil me han ayudado a poner en práctica y a renovar los conocimientos que tenía sobre esas dos tecnologías. Me ha gustado poder volver a tocarlas y poder llevar a cabo un proyecto real y de esta magnitud.

En definitiva, estoy muy contento del trabajo realizado. Es un proyecto que cuando me lo plantearon, sonaba muy bien y me gustó mucho la idea, pero que desarrollando el sistema me gustaba aún más, y ahora que veo los resultados, me encanta. Estoy seguro que seguiré avanzando en este mundo de la domótica y que seguiré tocando estas herramientas y tecnologías, ya sea en este o en otros proyectos.

Conocimientos de las asignaturas

Los conocimientos que he adquirido a lo largo de la carrera y que me han servido para desarrollar el proyecto son los siguientes:

- Arquitectura de Software (AS)
 - El diseño de arquitecturas me ha servido para diseñar el software.
- Aplicaciones y Servicios Web (ASW)
 - El diseño de servicios web me ha servido para diseñar el servicio que comunicará el Servidor.
- Diseño de Bases de Datos (DBD)
 - Me ha ayudado a realizar un diseño e implementar correctamente la base de datos.
- Ingeniería de Requisitos (ER)
 - Me ha ayudado en la asignatura de GEP, a identificar los objetivos del proyecto y las partes interesadas.
 - Me ha ayudado también a definir los requisitos y las historias de usuario.
- Gestión de Proyecto de Software (GPS)
 - Gracias a las técnicas aprendidas en GPS, he podido gestionar mejor el tiempo y tareas del proyecto, así como la visión global de éste.

- Proyecto de Ingeniería del Software (PES)
 - Como PES es un caso práctico de ejecución de un proyecto de software, gran parte de la asignatura ha sido de ayuda: planificación temporal, presupuesto, diseño del software, desarrollo y testing de éste.

Competencias técnicas del proyecto

Las competencias técnicas del proyecto que se han llevado a cabo son las siguientes:

- **CES1.1:** Desarrollar, mantener y evaluar sistemas y servicios software complejos i/o críticos. [En profundidad]
 - Esta competencia se ha completado con éxito ya que se ha desarrollado, mantenido y evaluado un gran sistema con una gran parte crítica. El sistema es crítico ya que toda la información tiene que estar disponible a tiempo real, y el sistema en sí no puede fallar en ningún momento ya que el cliente confía en el sistema para la protección del paciente.
- **CES1.2:** Dar solución a problemas de integración en función de las estrategias, de los estándares y de las tecnologías disponibles. [En profundidad]
 - Esta competencia se cumple ya que nuestro sistema integra los datos recogidos por los sensores en *Arduino* y los trata para poder enviarlos al servidor montado en la *Raspberry Pi*. La *Raspberry Pi* guarda los datos en una base de datos, los recupera en el servidor y los muestra en una página web y en una aplicación móvil.
- **CES1.8:** Desarrollar, mantener y evaluar sistemas de control y de tiempo real [Bastante]
 - El sistema es un sistema de control y de tiempo real debido a que toda la información está disponible al momento. Necesita ser así ya que hay que controlar los posibles problemas que surjan en los hogares de personas que necesitan una atención especial y actuar inmediatamente en caso de algún peligro. También el cliente requiere de nuestro sistema una actualización en tiempo real de los datos que visualiza desde la página web o el terminal móvil.
- **CES2.1:** Definir y gestionar los requisitos de un sistema software. [En profundidad]
 - Como el sistema parte de un concepto, hemos tenido que definir bien los requisitos del software. Se ha conseguido con una buena documentación de requisitos y reflejando estos en el sistema. Como es un sistema innovador y la empresa no tenía ningún documento ni se había planteado ninguna solución hasta el momento, los hemos tenido que definir y gestionar desde cero, por lo que el nivel de logro en esta competencia es de “en profundidad”.
- **CES2.2:** Diseñar soluciones apropiadas en uno o más dominios de aplicación, utilizando métodos de ingeniería de software que integren aspectos éticos, sociales, legales y económicos. [Bastante]
 - Se ha cumplido ya que el sistema integra aspectos éticos, sociales y económicos. Trata de una aplicación que mira por la seguridad y por el bienestar de las personas mayores o que necesiten una atención especial.

17. REFERENCIAS

- [1] FIB UPC (2015). Facultad de informática de Barcelona. < <http://www.fib.upc.edu> >
- [2] Worldline. < <http://es.worldline.com> >
- [3] Internet of Things Spain < <http://iot-spain.com> >
- [4] Lively < www.mylively.com >
- [5] PocketFinder < www.pocketfinder.com >
- [6] Guardian Medical Monitoring < www.guardianmedicalmonitoring.com >
- [7] MC10's BioSTAMP < www.mc10inc.com >
- [8] Independa < www.independa.com >
- [9] Teleasistencia < <http://www.cruzroja.es/principal/web/teleasistencia> >
- [10] Arduino < <https://www.arduino.cc/> >
- [11] BitBucket < <https://bitbucket.org/> >
- [12] Microsoft Lync < <https://www.microsoft.com/es-es/download/details.aspx?id=35451> >
- [13] Page Personnel < http://www.pagepersonnel.es/sites/pagepersonnel.es/files/er_tecnologia16.pdf >
- [14] Índice de variación mensual de los precios medios del gasóleo en España, Ministerio de Fomento
<
http://www.fomento.gob.es/mfom/lang_castellano/direcciones_generales/transporte_terrestre/servicios_transportista/inc_gasoleo_mensual.htm >
- [15] Especificaciones del Ford Fiesta < <http://www.km77.com/precios/ford/fiesta/2002/fiesta-5p-ambiente-14-tdci-68-cv> >
- [16] Transports Metropolitans de Barcelona < <http://www.tmb.cat/es/sistema-tarifari-integrat/-/ticket/TJove> >
- [17] One Plus < <https://oneplus.net/es/2> >
- [18] Historias de usuario < <https://www.mountangoatsoftware.com/agile/user-stories> >
- [19] INVEST < <http://jmbeas.es/guias/historias-de-usuario/> >
- [20] Raspberry Pi < <http://www.raspberrypi.org/> >
- [21] NOOBS < <https://www.raspberrypi.org/downloads/noobs/> >
- [22] Raspbian < <http://raspbian.org/> >
- [23] Debian < <https://www.debian.org/index.es.html> >
- [24] MQTT < <http://mqtt.org/> >
- [25] Mosquitto < <http://mosquitto.org/> >
- [26] openHAB < <http://www.openhab.org/> >
- [27] Arduino < <http://arduino.cl/que-es-arduino/> >
- [28] Placa Arduino UNO R3 < <http://arduino.cl/arduino-uno/> >
- [29] ZigBee < <http://www.digi.com/resources/standards-and-technologies/rfmodems/zigbee-wireless-standard> >
- [30] 802.15.4 < <http://standards.ieee.org/getieee802/download/802.15.4-2011.pdf> >
- [31] WPAN < <http://standards.ieee.org/about/get/802/802.15.html> >
- [32] ZigBee Alliance < <http://www.zigbee.org/> >
- [33] IEEE 802.11 < <http://standards.ieee.org/about/get/802/802.11.html> >
- [34] Wi-Fi Alliance < <http://www.wi-fi.org/> >
- [35] XBee < <http://www.digi.com/lp/xbee> >
- [36] Digi International < <http://www.digi.com/es/> >
- [37] ZigBee shield < <https://www.arduino.cc/en/Main/ArduinoXbeeShield> >
- [38] ESP8266 < <http://www.esp8266.com/> >
- [39] UART < <http://www.mioio.es/comunicacion-serie-uart/> >
- [40] My.openHAB < <https://my.openhab.org/> >
- [41] IFTTT < <https://ifttt.com/> >

18. BIBLIOGRAFIA

Instituto de Mayores y Servicios Sociales < <http://www.imserso.es/> >

Instituto Nacional de Estadística < <http://www.ine.es/> >

METODO EN CASCADA. En línea. < http://www.academia.edu/6362716/METODO_EN_CASCADA >

Sistema de control domótico. En línea < <https://riunet.upv.es/bitstream/handle/10251/18228/Memoria.pdf> >

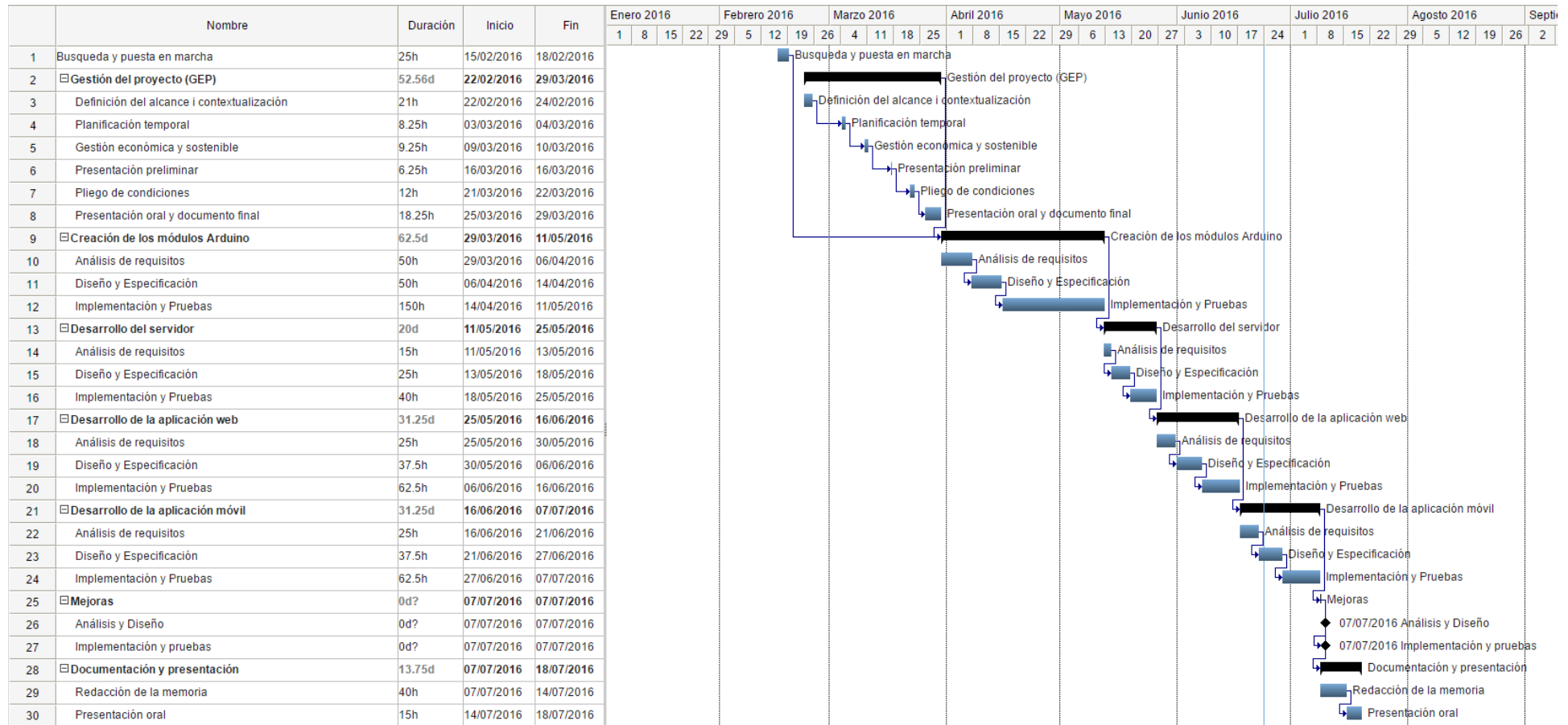
Red de sensores. En línea. <

<http://repositorio.upct.es/bitstream/handle/10317/3678/tfg176.pdf?sequence=1> >

GeekFactory, módulo WiFi Serial ESP8266 <

<http://www.geekfactory.mx/tienda/radiofrecuencia/modulo-wifi-serial-esp8266-economico/> >

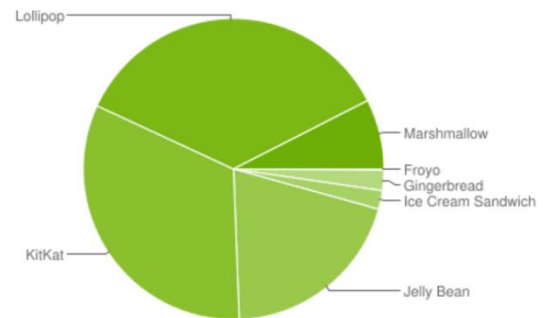
ANEXO 1. DIAGRAMA DE GANTT



ANEXO 2. VERSIONES DE ANDROID Y IOS

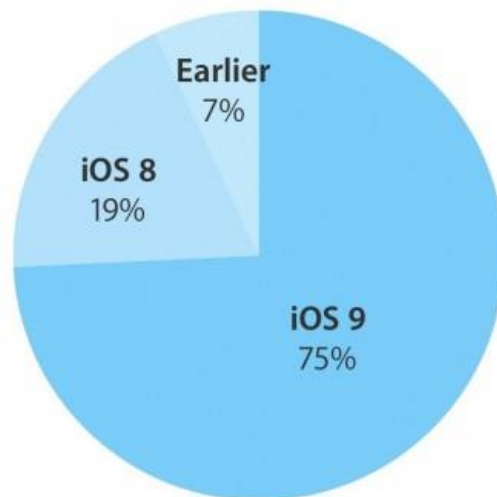
Versiones de Android utilizadas:

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.0%
4.1.x	Jelly Bean	16	7.2%
4.2.x		17	10.0%
4.3		18	2.9%
4.4	KitKat	19	32.5%
5.0	Lollipop	21	16.2%
5.1		22	19.4%
6.0	Marshmallow	23	7.5%



Versiones de iOS utilizadas:

75% of devices are using iOS 9.



As measured by the App Store on January 11, 2016.